# UM0834
# User manual

## Developing and debugging your STM8S-DISCOVERY application code

## Introduction

This document complements the information in the STM8S datasheets by describing the software environment and development recommendations required to build an application around the STM8S-DISCOVERY. It also explains how to use the STM8S firmware and STM8S touch sensing libraries provided by STMicroelectronics, in order to develop cost-effective applications.

In addition, ST provides a development package which can be used to build an application running on the STM8S-DISCOVERY. This package includes application code examples and a project template.

## Reference documents

- STM8S-DISCOVERY user manual (UM0817)
- ST Visual Develop (STVD) user manual (UM0036)
- Adjustable LED blinking speed using STM8S-DISCOVERY touch sensing key (UM0833)
- Displaying variable voltage on a LED bar using STM8S-DISCOVERY (UM0848)
- Generating PWM signals using STM8S-DISCOVERY (UM0856)
- STM8S reference manual (RM0016)
- STM8S105xx datasheet

# Contents


**1 Overview of STM8 software development toolchain . . . . . 6**

1.1 ST Visual Develop (STVD) . . . . . 6

1.2 ST Visual Programmer (STVP) . . . . . 7

1.3 C and assembly compilers . . . . . 7

1.4 Firmware libraries . . . . . 7

**2 Installing the development toolchain . . . . . 8**

2.1 Downloading and installing STVD . . . . . 8

2.2 Downloading and installing the compilers . . . . . 8

2.2.1 Installing the Cosmic compiler . . . . . 8

2.2.2 Installing the Raisonance compiler . . . . . 11

2.2.3 Compiler settings . . . . . 14

**3 Description of firmware libraries . . . . . 15**

3.1 STM8S standard firmware library . . . . . 15

3.1.1 Description of the STM8S standard firmware library . . . . . 16

3.1.2 STM8S standard firmware library online help . . . . . 17

3.2 STM8S touch sensing library . . . . . 18

3.2.1 STM8S touch sensing library working principles . . . . . 18

3.2.2 Configuring the STM8S touch sensing library . . . . . 19

3.2.3 Developing your application with the STM8S touch sensing library . . . 20

3.2.4 STM8S touch sensing library online help . . . . . 21

**4 Configuring the option bytes for your application . . . . . 22**

**5 Building and running your STVD project . . . . . 25**

5.1 **Description of the STM8S-DISCOVERY development package . . . . . 25**

5.1.1 Libraries . . . . . 25

5.1.2 Application code examples . . . . . 26

5.1.3 Project structure . . . . . 26

5.2 Running the STM8S-Discovery examples . . . . . 27

5.3 Creating your STVD project . . . . . 28

5.3.1 Creating your project structure . . . . . 28

5.3.2 Creating your STVD project . . . . . 29
5.3.3 Linking the libraries to your STVD project . . . . . 31
5.4 Updating the libraries . . . . . 36
5.4.1 Updating the STM8S standard firmware library . . . . . 36
5.4.2 Updating the STM8 touch sensing library . . . . . 36

**6 Building, debugging and running your application . . . . . 37**
6.1 Building your application . . . . . 37
6.2 Debugging your application . . . . . 38
6.3 Online help . . . . . 40

**7 Revision history . . . . . 41**

# List of tables

Table 1. Touch sensing library configuration for STM8S-DISCOVERY using TS1 touchkey . . . . . 20
Table 2. Document revision history . . . . . 41

# List of figures

Figure 1. STVD overview . . . 6
Figure 2. Cosmic compiler form . . . 9
Figure 3. Downloading Cosmic compiler software wizard . . . 10
Figure 4. Cosmic compiler installation wizard . . . 10
Figure 5. Cosmic compiler . . . 11
Figure 6. Raisonance compiler installation wizard . . . 12
Figure 7. Activation code registration . . . 13
Figure 8. Personal information form . . . 13
Figure 9. Serial key information form . . . 14
Figure 10. Selecting the Cosmic compiler for the Discover project . . . 14
Figure 11. STM8S standard firmware library architecture . . . 16
Figure 12. stm8s_conf.h peripheral define statements . . . 17
Figure 13. STM8 firmware library online help home page . . . 17
Figure 14. STM8S touch sensing library software architecture . . . 18
Figure 15. STM8S touch sensing functional architecture . . . 19
Figure 16. User application flowchart . . . 21
Figure 17. STM8S touch sensing library online help home page . . . 21
Figure 18. Select your MCU . . . 22
Figure 19. STVP option byte selection menu . . . 23
Figure 20. STVP option byte programming menu . . . 24
Figure 21. STVP option byte programming message . . . 24
Figure 22. STM8S-Discovery_dev package content . . . 25
Figure 23. Selecting the debug instrument . . . 28
Figure 24. Creating your project structure . . . 28
Figure 25. Creating the STVD workspace for your project . . . 29
Figure 26. Checking the selected compiler . . . 30
Figure 27. Select the MCU . . . 30
Figure 28. Copy stm8s.h to Include Files/FWLib . . . 31
Figure 29. No libraries linked . . . 32
Figure 30. Linking the STM8S standard firmware library - step 1 . . . 32
Figure 31. Linking the STM8S standard firmware library - step 2 . . . 33
Figure 32. Linking the STM8 touch sensing library - step 1 . . . 35
Figure 33. Linking the STM8 touch sensing library - step 2 . . . 36
Figure 34. Selecting the project configuration . . . 37
Figure 35. Building your project . . . 38
Figure 36. Building successful message . . . 38
Figure 37. Debugging your application . . . 39
Figure 38. Peripheral registers window . . . 39
Figure 39. Watch window . . . 40
Figure 40. STVD online help . . . 40

# 1 Overview of STM8 software development toolchain

To develop, compile and run an application software on an STM8S microcontroller, the following software toolchain components are required:

- Integrated development environment (IDE) composed of the ST Visual Develop (STVD) and the ST Visual Programmer software interface (STVP)
- Compilers
- Firmware libraries: they are optional, and allow to easily create a new application

STMicroelectronics provides a free software package including STVD and STVP.

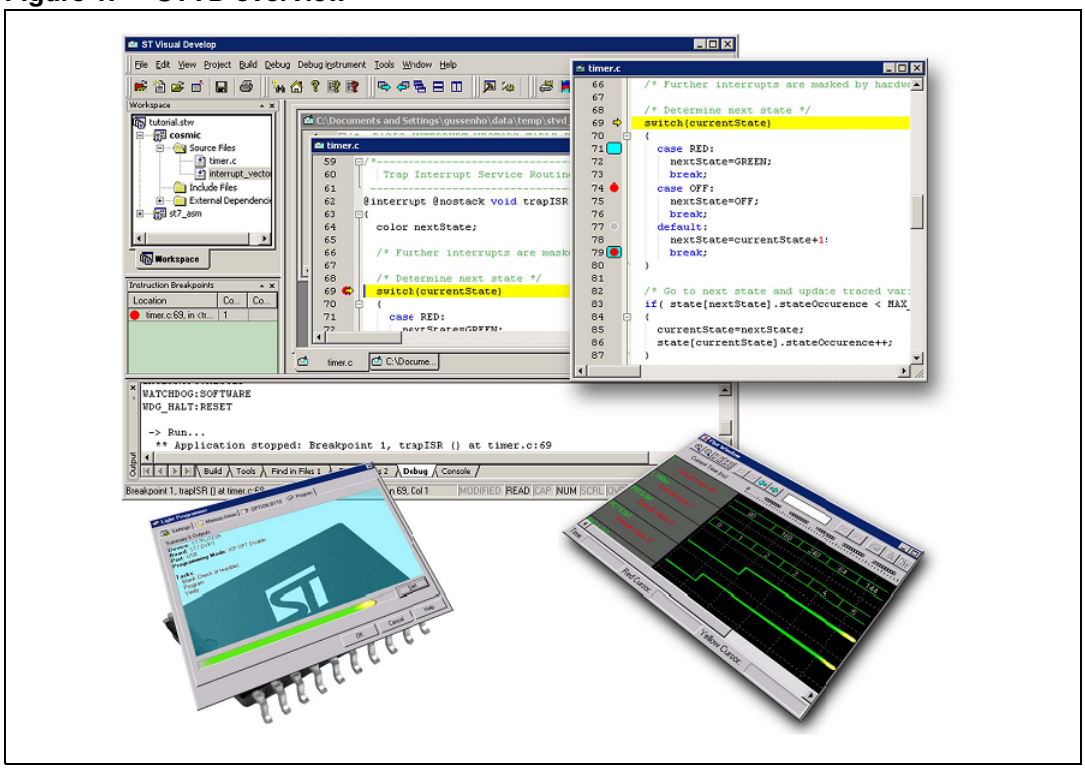
## 1.1 ST Visual Develop (STVD)

STVD is a full-featured development environment. It is a seamless integration of the Cosmic and Raisonance C compilers for STM8 microcontroller family. These compilers are free when developing code up to 16 Kbytes.

STVD main features are:

- Seamless integration of C and ASM compilers
- Full-featured debugger
- Project management
- Syntax highlighting editor
- Integrated programming interface

**Figure 1. STVD overview**



## 1.2 ST Visual Programmer (STVP)

STVP is a easy-to-use graphical interface allowing to read, write and verify the code and data programmed in your STM8 microcontroller Flash program memory, data EEPROM and option bytes. STVP also features a project mode for saving programming configurations and automating programming sequences.

## 1.3 C and assembly compilers

The C and assembly compilers are seamlessly integrated into the STVD development environment. They allow to directly configure and control the building of your application from an easy-to-use graphical interface.

The supported compilers are the following:

- Cosmic C compiler for STM8 (free version up to 16 Kbytes of code)

  For more information, refer to http://www.cosmic-software.com.
- Raisonance C compiler for STM8 (free up to 16 Kbytes of code)

  For more information, refer to http://www.raisonance.com.
- STM8 assembler linker

  This is a free assembly toolchain included in the STVD toolset. It allows to assemble and link your application source code.

## 1.4 Firmware libraries

The STM8S standard firmware library is a complete package consisting of drivers for all the standard peripherals of Performance line STM8S20x and Access line STM8S10x microcontrollers. It is written in strict ANSI-C code and is fully MISRA C 2004 compliant.

The STM8S touch sensing library endows any STM8 microcontroller with touch sensing capabilities. This firmware offers a complete and robust solution to manage capacitive sensing keys, wheels, and sliders.

Refer to *Section 3.1: STM8S standard firmware library* and *Section 3.2: STM8S touch sensing library* for details on the installation and configuration of these libraries.

# 2 Installing the development toolchain

## 2.1 Downloading and installing STVD

STVD software is available from http://www.st.com.

To install STVD, download the installation software and follow each step of the installation wizard.

When the installation is complete, the executable is available from ***START>Programs>ST Toolset>Developement Tools>ST Visual Develop*** or under **C:\Program Files\STMicroelectronics\st_toolset\stvd**.

## 2.2 Downloading and installing the compilers

Cosmic and Raisonance compilers are compatible with STVD. They are available together with their documentation at the following urls:

- Cosmic: http://www.cosmicsoftware.com/download_stm8_16k.php
- Raisonance: http://www.mcu-raisonance.com

*Note: A free license is required to use the compilers.*

### 2.2.1 Installing the Cosmic compiler

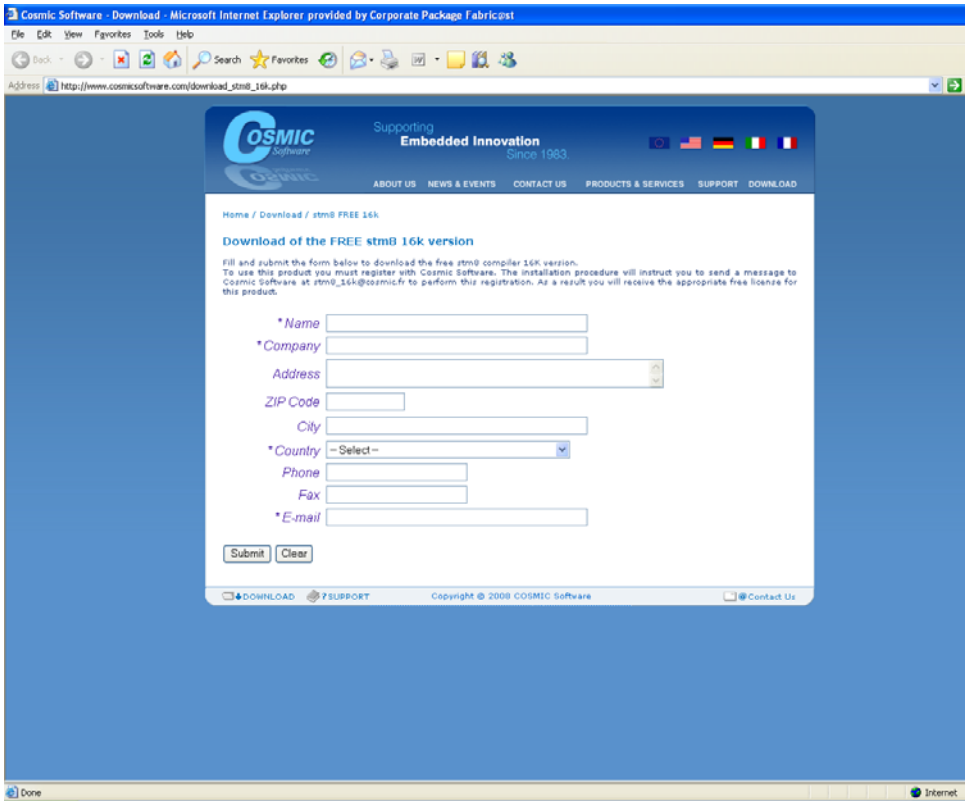
To install the Cosmic compiler, follow the sequence described below:

1. Step 1

   Connect to http://www.cosmicsoftware.com/download_stm8_16k.php, and fill in the information form (see *Figure 2*).

   Click **Submit** to access the download page.

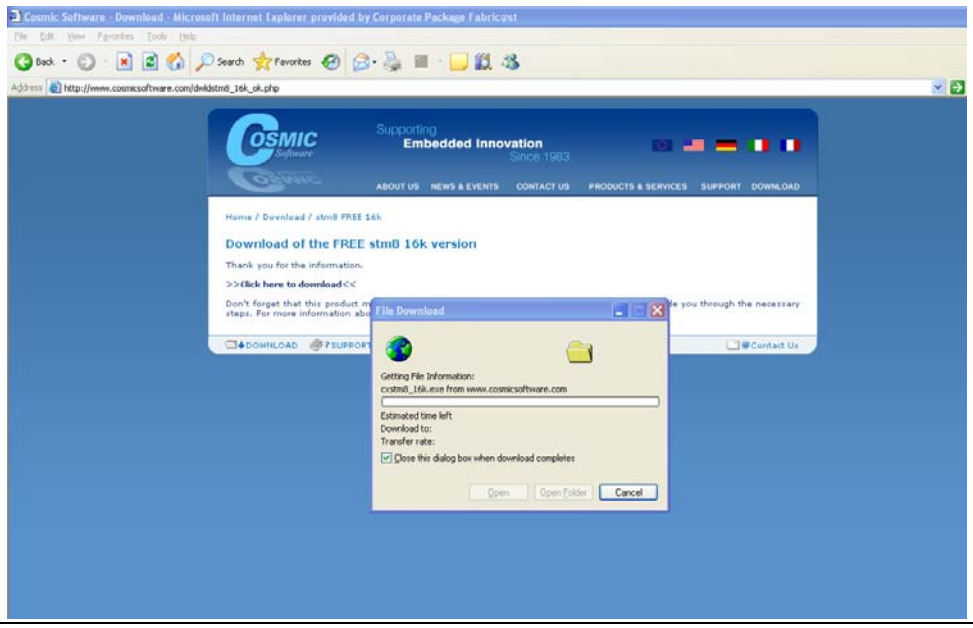
**Figure 2. Cosmic compiler form**


Cosmic Software - Download - Microsoft Internet Explorer provided by Corporate Package Fabrica
File Edit View Favorites Tools Help
Address http://www.cosmicsoftware.com/download_stm8_16k.php
COSMIC Software
Supporting
Embedded Innovation
Since 1983
ABOUT US NEWS & EVENTS CONTACT US PRODUCTS & SERVICES SUPPORT DOWNLOAD
Home / Download / stm8 FREE 16k
Download of the FREE stm8 16k version
Fill and submit the form below to download the free stm8 compiler 16K version.
To use this product you must register with Cosmic Software. The installation procedure will instruct you to send a message to Cosmic Software at stm8_16k@cosmic.fr to perform this registration. As a result you will receive the appropriate free license for this product.
*Name
*Company
Address
ZIP Code
City
*Country
– Select –
Phone
Fax
*E-mail
Submit
Clear
DOWNLOAD
SUPPORT
Copyright © 2008 COSMIC Software
Contact Us
Done
Internet

2. Step 2

   Download and run the installation software wizard (see *Figure 3*).

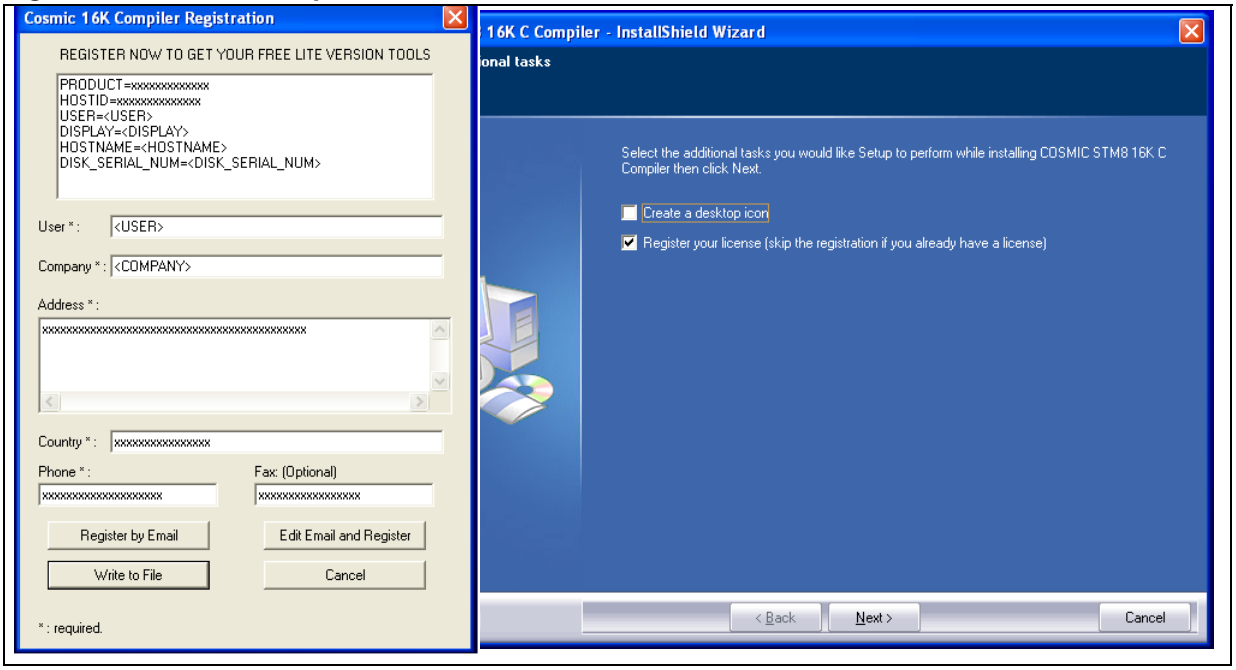
**Figure 3. Downloading Cosmic compiler software wizard**




3. Step 3

   Follow the wizard instructions to install the Compiler (see *Figure 4*). Do not forget to register for a free license and click **Register by Email**. You will receive the license file by e-mail.

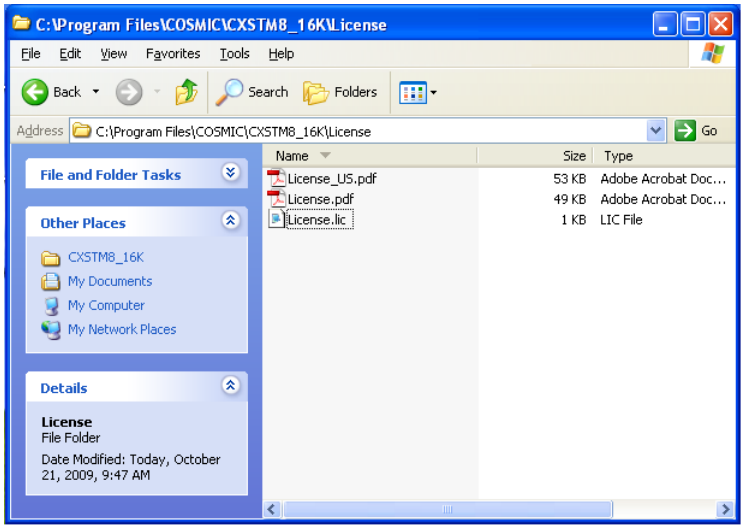
**Figure 4. Cosmic compiler installation wizard**



4. Step 4

   Copy the licence file that you have received into the directory ***\Program files\COSMIC\CXSTM8_16K\License*** *(see Figure 5).*

**Figure 5. Cosmic compiler**



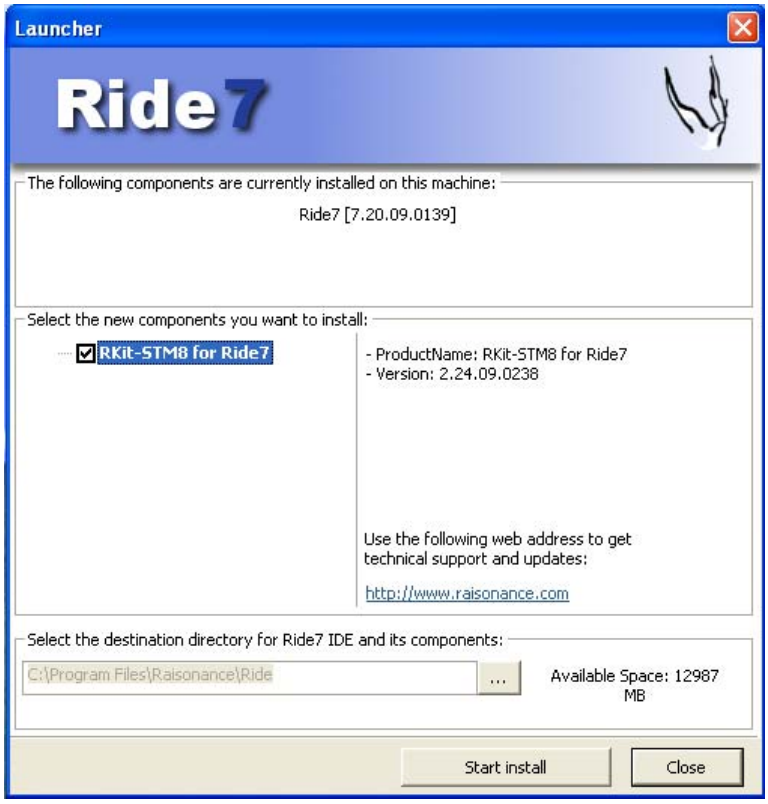

## 2.2.2 Installing the Raisonance compiler

To install the Raisonance compiler, follow the sequence described below:

1. Step 1

   Download Ride and the Raisonance kit RKit-STM8 from http://www.raisonance-mcu.com, and follow the installation wizard instructions (see *Figure 6*).

**Figure 6. Raisonance compiler installation wizard**


Launcher
Ride7
The following components are currently installed on this machine:
Ride7 [7.20.09.0139]
Select the new components you want to install:
RKit-STM8 for Ride7
- ProductName: RKit-STM8 for Ride7
- Version: 2.24.09.0238
Use the following web address to get technical support and updates:
http://www.raisonance.com
Select the destination directory for Ride7 IDE and its components:
C:\Program Files\Raisonance\Ride
...
Available Space: 12987 MB
Start install
Close

2. Step 2 - Register for a free 16 Kbyte license

   a) You need a computer serial number. To get it, open Ride, click **Help**>**Licence**, and enter your name, you company, and select **Manual installation**. The serial key will appear as shown in *Figure 7*.

   b) To get your free 16 Kbyte activation code, go to http://www.raisonance-mcu.com. Fill the question form with your personal information (see *Figure 8*), and the registration form with your serial key (see *Figure 9*).

   After this operation, you will receive your activation code by e-mail, and you will be granted a free 16 Kbyte Raisonance compiler licence to be used by STVD.

**Figure 7. Activation code registration**

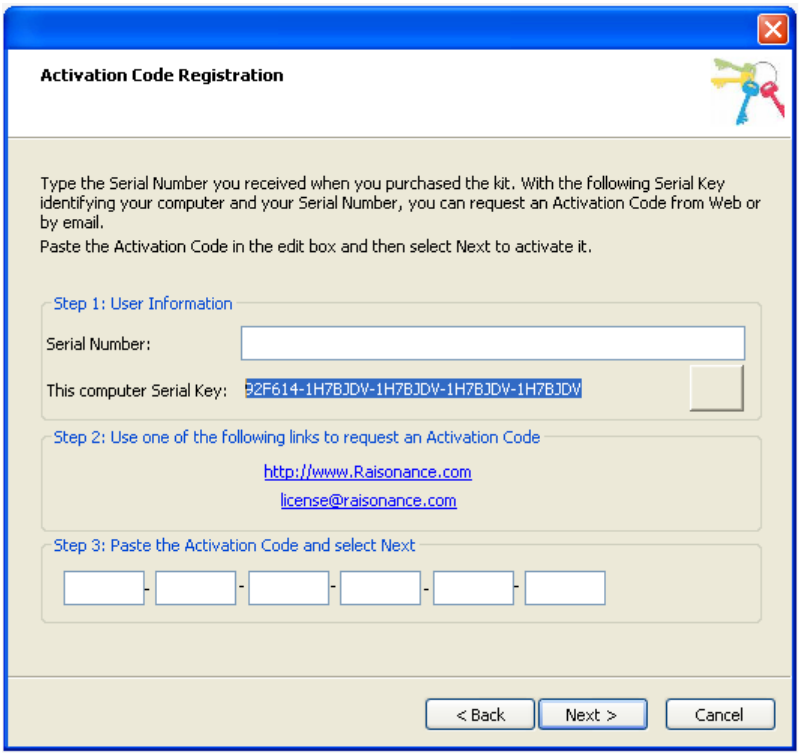

Activation Code Registration
Type the Serial Number you received when you purchased the kit. With the following Serial Key identifying your computer and your Serial Number, you can request an Activation Code from Web or by email.
Paste the Activation Code in the edit box and then select Next to activate it.
Step 1: User Information
Serial Number:
This computer Serial Key:
92F614-1H7BJDV-1H7BJDV-1H7BJDV-1H7BJDV
Step 2: Use one of the following links to request an Activation Code
http://www.Raisonance.com
license@raisonance.com
Step 3: Paste the Activation Code and select Next
< Back
Next >
Cancel

**Figure 8. Personal information form**

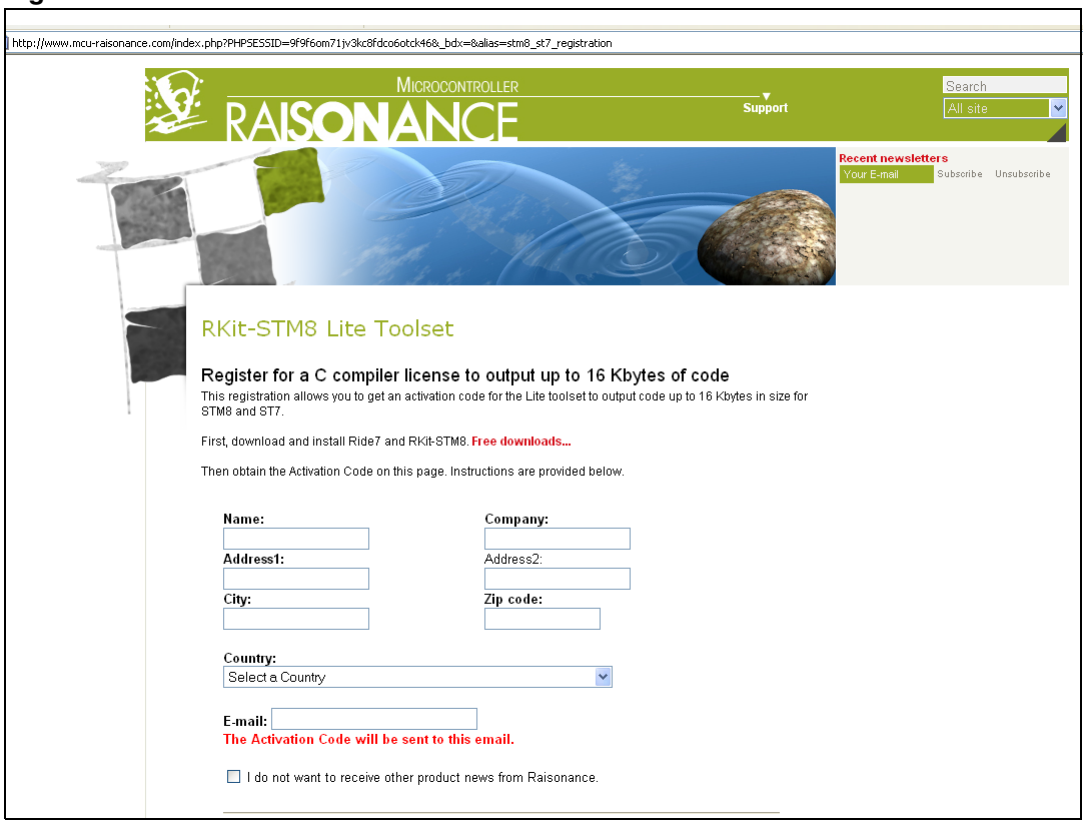
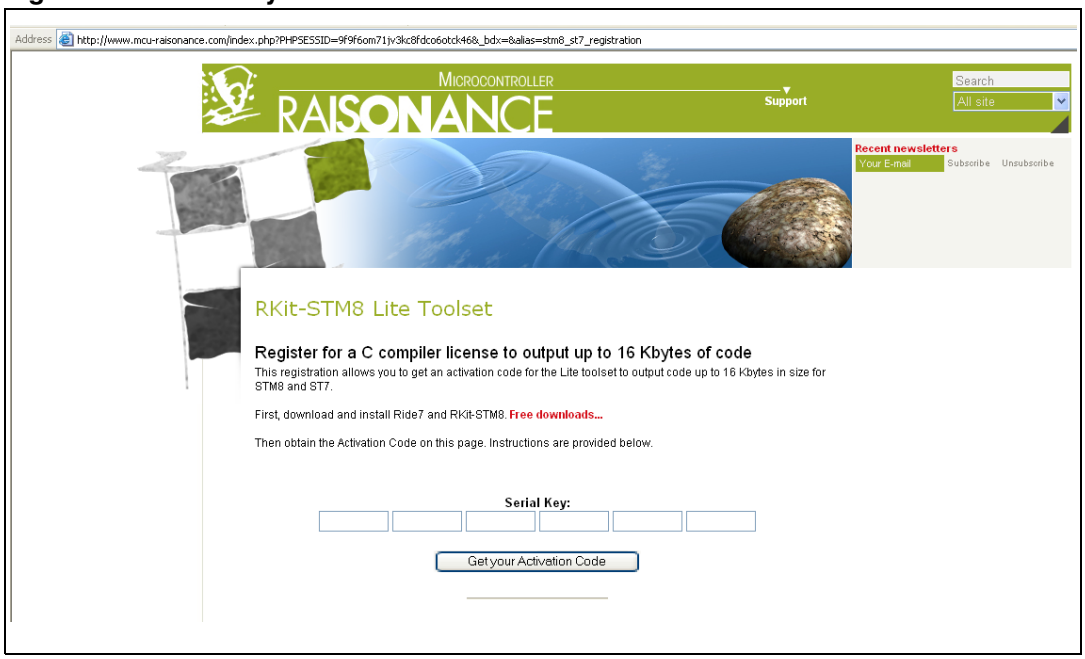

http://www.mcu-raisonance.com/index.php?PHPSESSID=9f9f6om71jv3kc8fdco6otck46&_bdx=8alias=stm8_st7_registration
MICROCONTROLLER
RAISONANCE
Support
Search
All site
Recent newsletters
Your E-mail
Subscribe
Unsubscribe
RKit-STM8 Lite Toolset
Register for a C compiler license to output up to 16 Kbytes of code
This registration allows you to get an activation code for the Lite toolset to output code up to 16 Kbytes in size for STM8 and ST7.
First, download and install Ride7 and RKit-STM8. Free downloads...
Then obtain the Activation Code on this page. Instructions are provided below.
Name:
Company:
Address1:
Address2:
City:
Zip code:
Country:
Select a Country
E-mail:
The Activation Code will be sent to this email.
I do not want to receive other product news from Raisonance.

**Figure 9. Serial key information form**

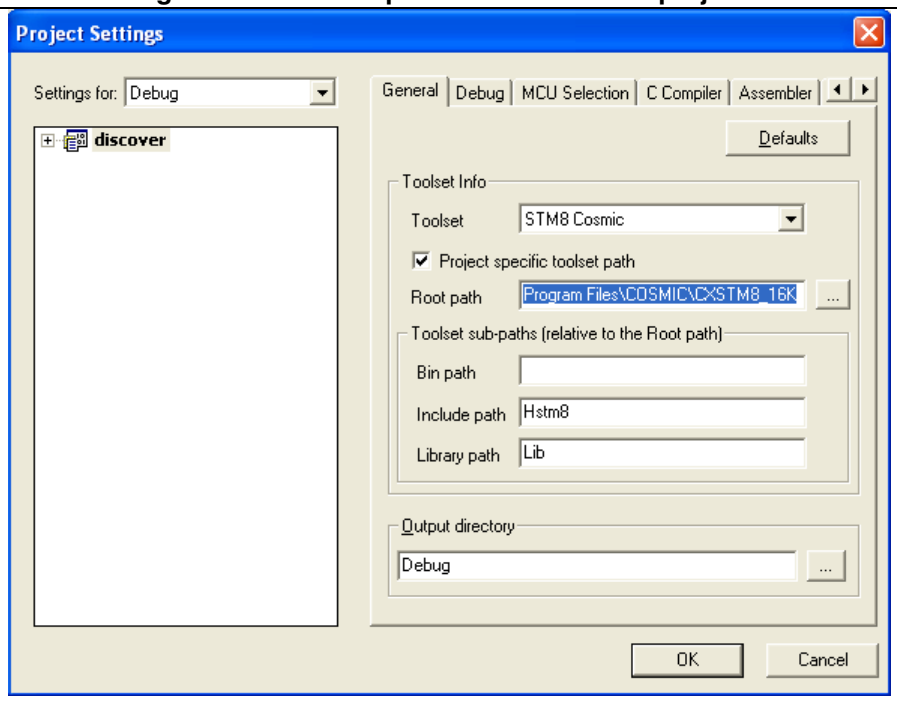

Address
http://www.mcu-raisonance.com/index.php?PHPSESSID=9f9f6om71jv3kc8fdco6otck46&_bdx=&alias=stm8_st7_registration
MICROCONTROLLER
RAISONANCE
Support
Search
All site
Recent newsletters
Your E-mail
Subscribe
Unsubscribe
RKit-STM8 Lite Toolset
Register for a C compiler license to output up to 16 Kbytes of code
This registration allows you to get an activation code for the Lite toolset to output code up to 16 Kbytes in size for STM8 and ST7.
First, download and install Ride7 and RKit-STM8. Free downloads...
Then obtain the Activation Code on this page. Instructions are provided below.
Serial Key:
Get your Activation Code

## 2.2.3 Compiler settings

STVD uses the default compiler defined at the first launch of the toolchain. This compiler is not defined during the installation.

You can select a different compiler for a specific project in the **General tab** of the STVD **Project**>**Settings** window.

**Figure 10. Selecting the Cosmic compiler for the Discover project**



# 3 Description of firmware libraries

The STM8S microcontroller family is provided with both the STM8S standard firmware library and the touch sensing firmware library.

It is strongly recommended to use the libraries to develop your project as proposed in the the STM8S-Discovery development package and implemented in the Discover and PWM projects (see *Section 5.1: Description of the STM8S-DISCOVERY development package*).

The development package also contains the ADC project which uses neither the STM8S standard firmware library nor the touch sensing firmware library.

The STM8S standard firmware library and the STM8S touch sensing library are part of the STM8S-DISCOVERY development package that is available from the STM8S-DISCOVERY web page at http://www.st.com/stm8s-discovery.

**Warning: The STM8S touch sensing library includes the STM8S standard firmware library. Make sure you have the latest version of each library (see http://www.st.com/mcu).**

## 3.1 STM8S standard firmware library

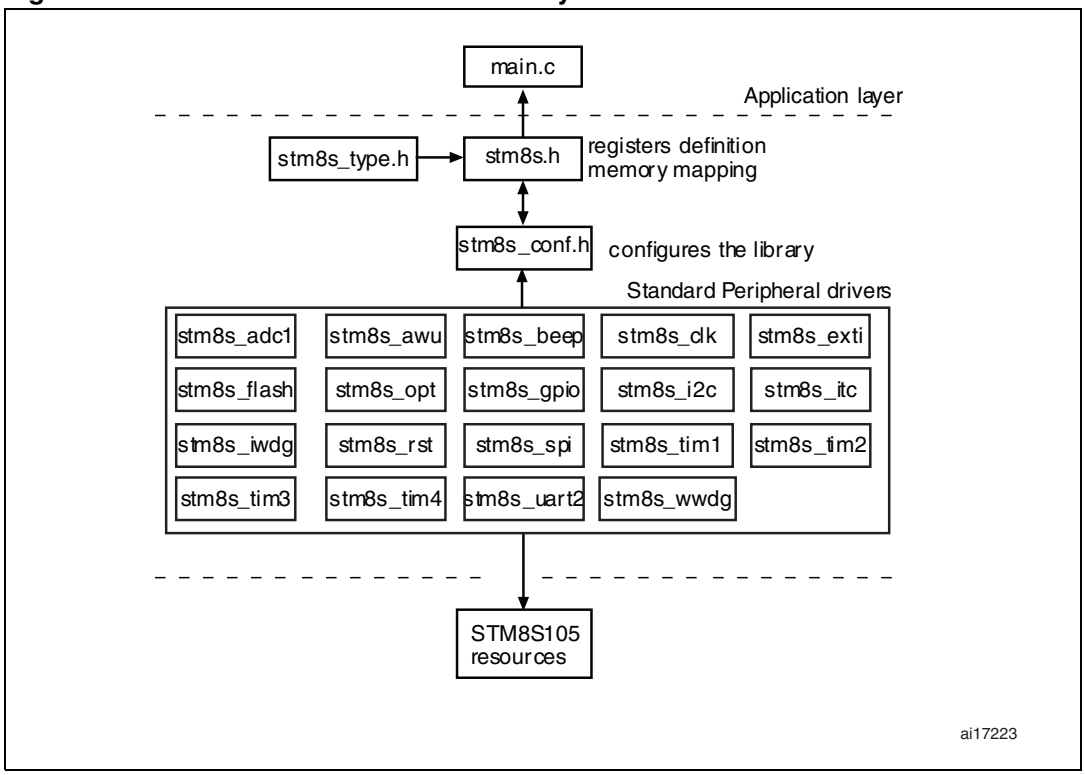
The STM8S standard firmware library contains a collection of routines, data structures, and macros covering the features of the STM8S peripherals, as well as a description of the device drivers (see *Figure 11*).

The STM8S standard firmware library allows to develop an application on any STM8S device without the need for in-depth study of each peripheral specifications. It saves significant time that would otherwise be spent in coding, while simultaneously reducing application development and integration cost.

It contains a set of example for each peripheral. All these examples are provided with workspace and project definition files for STVD and Cosmic C compiler to allow loading and compiling them easily into you development environment. These examples are developed to run on an STM8S208xx device with STMicroelectronics STM8/128-EVAL evaluation board.

They have to be considered as example codes to develop your own application. Some examples have already been tailored to run on the STM8S-DISCOVERY. They are available from the STM8S-DISCOVERY web pages at http://www.st.com/stm8s-discovery.

**Figure 11. STM8S standard firmware library architecture**




### 3.1.1 Description of the STM8S standard firmware library

The *stm8s.h* header file contains the definitions of constants and register structures for all peripherals. Uncomment `#define USE_STDPERIPH_DRIVER` when using the STM8S standard firmware library.

In addition, stm8s.h must be included in your main() routine.

The *stm8s_conf.h* file of the STM8S standard firmware library is used to configure the library by enabling the peripheral functions that are only used by your application (see *Figure 12*). In *stm8s_conf.h*, some peripheral define statements are conditioned by supported devices. For example:

```
#if defined(STM8S208) || defined(STM8S207) || defined(STM8S105)
#define _TIM3 (1)
#endif
```

In *stm8s_conf.h*, the HSE value define statement may be adjusted to the oscillator frequency or to the external clock generator frequency. It is also conditioned by supported devices. Make sure you have the correct value for the STM8S-DISCOVERY external oscillator (expressed in Hz). For example:

```
#define HSE_VALUE ((u32)16000000)
```

The peripheral interrupt function file, stm8s_it.c, must be modified to include the code to handle the interrupts used by your application. The *stm8s_type.h* file includes common

types and constants used by the peripheral drivers. Each peripheral driver is made up of the following files:

- The source code *stm8s_<periph>.c* containing all the software functions required to use the corresponding peripheral.
- The header *stm8s_<periph>.h* including the peripheral function prototypes as well as the variables, constant and structures used within these functions.

The flow that must be followed to create your application software using the STM8S standard firmware library is described in *Section 5.3: Creating your STVD project*.

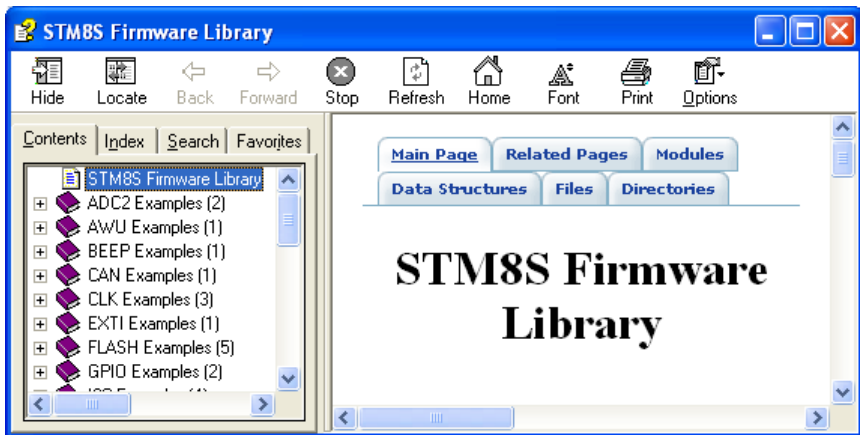
**Figure 12. *stm8s_conf.h* peripheral define statements**

```
/************************************* CLK ***********************
 #define _CLK (1)

/************************************* EXTI **********************
/* #define _EXTI (1) */

/******************************** FLASH/DATA EEPROM **************
/* #define _FLASH (1) */

/******************************** OPTION BYTES *******************
/* #define _OPT (1) */

/************************************* GPIO **********************
#define _GPIO (1)
```

### 3.1.2 STM8S standard firmware library online help

An online help, *stm8s_fwlib_um.chm,* is available inside the firmware installation directory to help you with the structure of the library (see *Figure 13*).

**Figure 13. STM8 firmware library online help home page**



## 3.2 STM8S touch sensing library

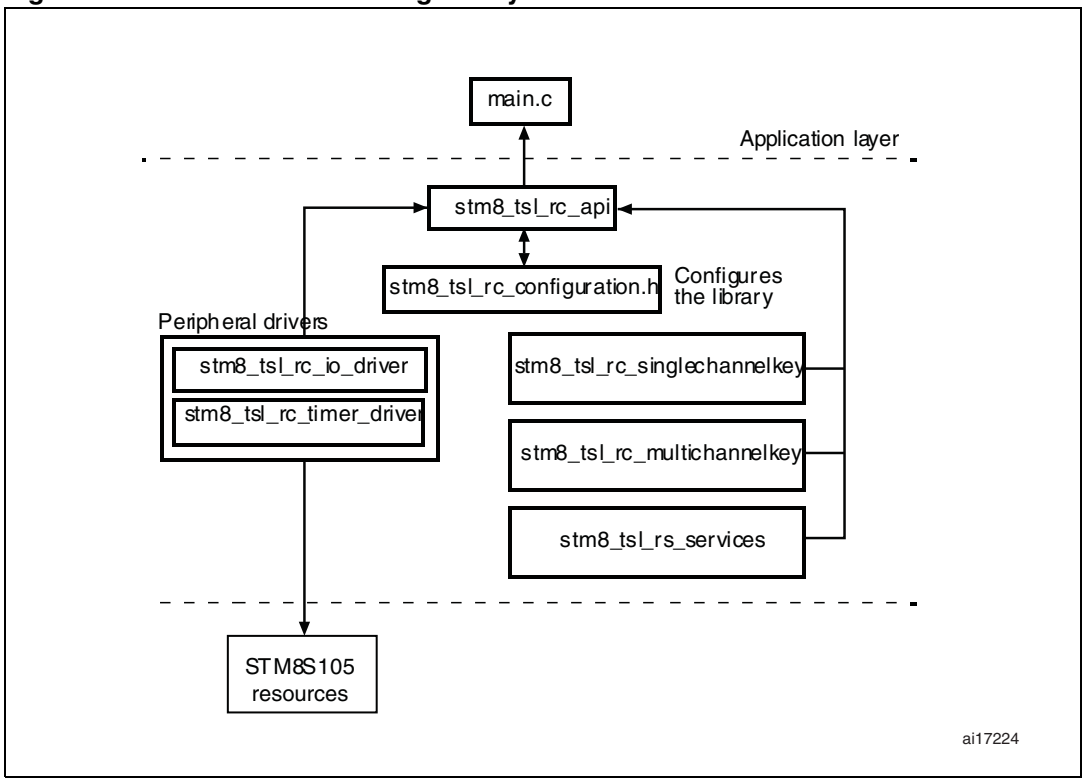
The STM8S touch sensing library is composed of a set of functions, variables and structures (see *Figure 14*).

The STM8S touch sensing library allows detecting physical touch on the capacitive sensing keys by controlling the charge/discharge timing cycle of an RC network composed of a single resistor and the touch electrode capacitance. Any variation in the RC charge/discharge timing resulting from a change in the electrode capacitance is detected and then filtered.

For detailed information related to the RC acquisition principle, refer to the application note AN2927 available on http://www.st.com/mcu.

**Warning: The STM8S touch sensing library versions earlier than V2.X.X are not compatible with the Raisonance compiler.**

**Figure 14. STM8S touch sensing library software architecture**




### 3.2.1 STM8S touch sensing library working principles

The STM8S touch sensing library is a set of compatible C files. It includes the API that makes the interface with the other software layers, a core set of files performing basic functions (main state machine, key state machine, services), plus the drivers to control timers and GPIOs.

The touch sensing library is based on a 2 state machines (see *Figure 15*):

- Main state machine

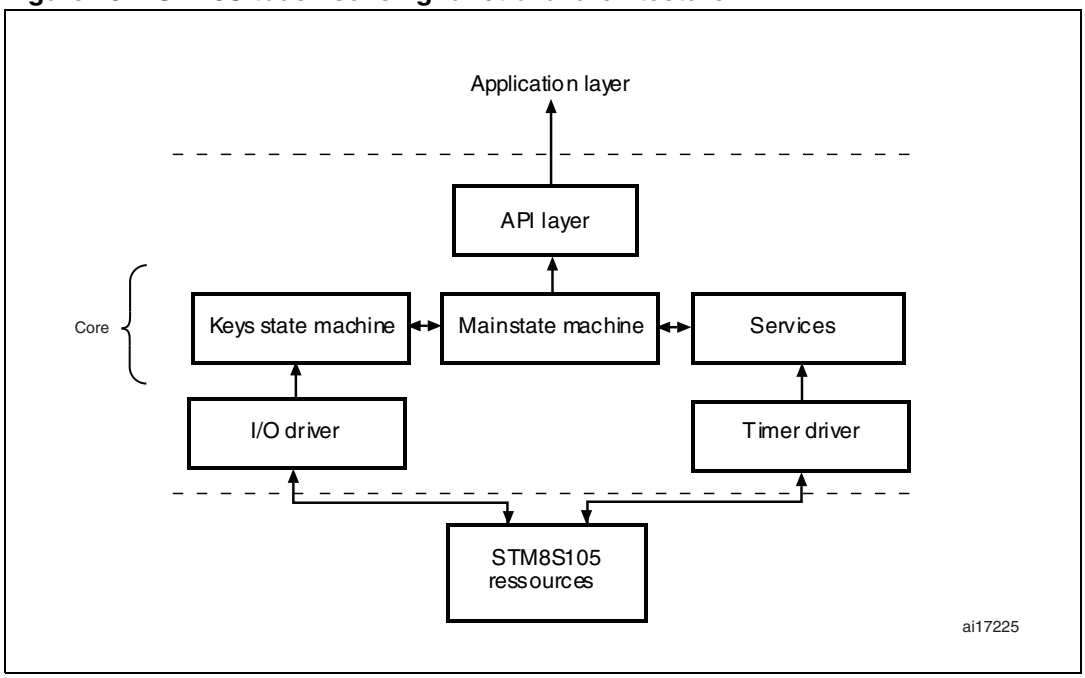
  The main state machine manages the sequence of the common actions concerning the touch sensing keys:

  – Signal acquisition
  – Data interpretation

- Key state machine

  Each key has its own state machine to manage its states: calibration, idle, detect, etc...

**Figure 15. STM8S touch sensing functional architecture**




## 3.2.2 Configuring the STM8S touch sensing library

In applications using the STM8S touch sensing library, all the library sources and header files must be added to the project.

*stm8_tsl_rc_api.c* defines the functions used to communicate between the STM8S touch sensing library and the user code. It is associated with *stm8_tsl_rc_api.h* that contains the API function prototypes, structures, variables and constants. *stm8_tsl_rc_api.h* must be included in the main() routine.

*STM8_TSL_RC_Configuration.h* contains the library static configuration parameters that must be set in compliance with your application hardware. Check that the define statements are filled in with the correct values.

See *Table 1* for an example of define statement configuration for an application using the single touch sensing button of STM8S-DISCOVERY (TS1).

The other define statements must remain unchanged (thresholds, integrator settings, IIR filter settings,...).

*Note: The STM8_TSL_RC_Configuration.h file can be updated to support up to 24 keys and 2 wheels or sliders.*

**Table 1. Touch sensing library configuration for STM8S-DISCOVERY using TS1 touchkey**

| Function | #define statement | Value | Comment |
|---|---|---|---|
| MCU selection | STM8S | 1 | - |
| Acquisition timer | TIMACQ<br>TIMACQ_CNTR_ADD | TIM3<br>0x5328 | TIM3 base address |
| Time-base timer | TIMTICK | TIM4 | - |
| Load I/O | LOADREF_PORT_ADDR<br>LOADREF_BIT | GPIOC_BaseAdress<br>0x04 | Port PC4 selected |
| Single channel key | SCKEY_P1_KEY_COUNT<br>SCKEY_P1_PORT_ADDR<br>SCKEY_P1_A<br>SCKEY_P1_DRIVEN_SHIELD_MASK<br>SCKEY_P2_COUNT<br>SCKEY_P3_COUNT | 1<br>GPIOC_BaseAddress<br>0x02<br>0x08<br>0<br>0 | Number of keys = 1<br>Port PC selected<br>Pin 1 selected as acquisition input<br>Pin 3 for active shield<br>Key port P2 not used<br>Key port P3 not used |
| Multichannel key | NUMBER_OF_MULTI_CHANNEL_KEYS | 0 | Multichannel key feature disabled |
| Electrode mask | GPIOA_ELECTRODES_MASK<br>GPIOB_ELECTRODES_MASK<br>GPIOC_ELECTRODES_MASK<br>GPIOD_ELECTRODES_MASK<br>GPIOE_ELECTRODES_MASK<br>GPIOF_ELECTRODES_MASK<br>GPIOG_ELECTRODES_MASK<br>GPIOH_ELECTRODES_MASK<br>GPIOI_ELECTRODES_MASK | 0x00<br>0x00<br>0x0A<br>0x00<br>0x00<br>0x00<br>0x00<br>0x00<br>0x00 | Defines the electrode mask for each GPIO used.<br>Mask must be set to 0x00 for unused GPIOs. |

### 3.2.3 Developing your application with the STM8S touch sensing library

*Figure 16* shows the main flowchart that is to be followed to develop an application using the touch sensing library with the STM8S-DISCOVERY. The application code must call some API functions to initialize the library and manage the acquisition state machines.

The function *TSL_Init()* must be called once during the initialization of the complete system. It is usually done in the main routine.

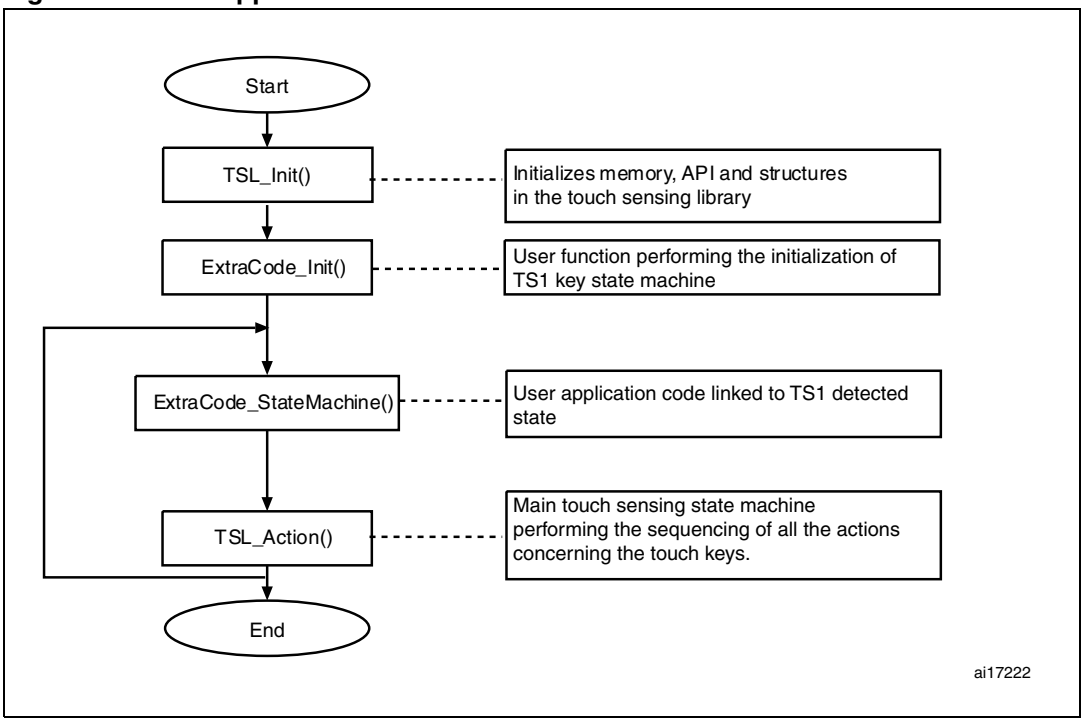
The *ExtraCode_Init()* performs the initialization of all the touchkeys implemented and enabled in the *STM8_TSL_RC_Configuration.h*.

The function *TSL_Action()* must be called periodically during system execution. It is the main state machine which sequences all the actions concerning all the touchkeys.

The application user code must be developed inside the function *ExtraCode_StateMachine()*. As an example, download the Discover application demonstration available at http://www.st.com/stm8s-discovery.

To read the data set by the library (for example to check flags), ensure that the touch sensing main state machine is in idle state to avoid spurious values to be retrieved.

**Figure 16. User application flowchart**


Start
TSL_Init()
Initializes memory, API and structures in the touch sensing library
ExtraCode_Init()
User function performing the initialization of TS1 key state machine
ExtraCode_StateMachine()
User application code linked to TS1 detected state
TSL_Action()
Main touch sensing state machine performing the sequencing of all the actions concerning the touch keys.
End
ai17222

## 3.2.4 STM8S touch sensing library online help

An online help, *stm8_tsl_um.chm*, is located in the STM8S touch sensing library installation directory to help you with the structure of the library (see *Figure 17*).

**Figure 17. STM8S touch sensing library online help home page**

# 4 Configuring the option bytes for your application

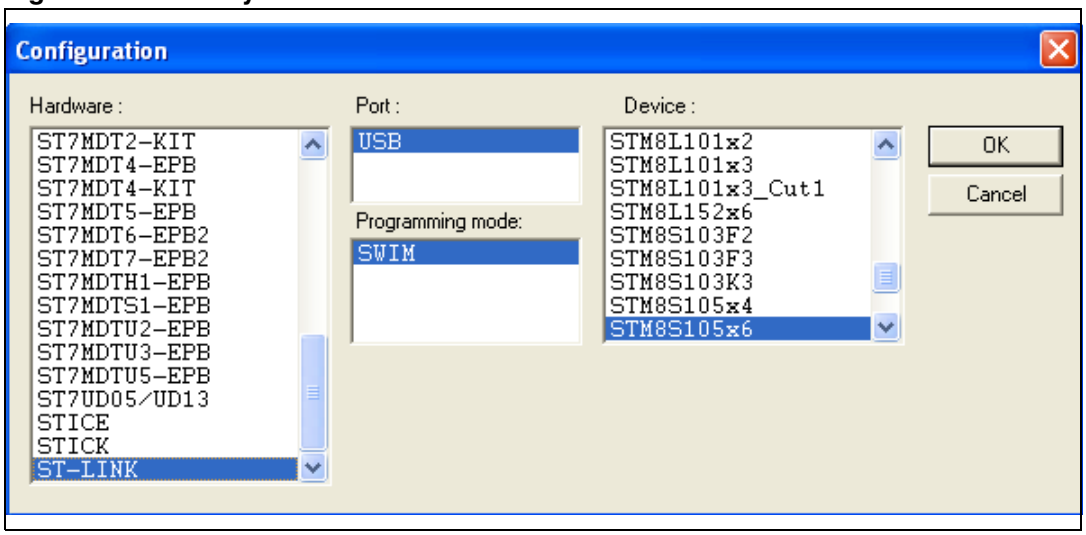
The option bytes allow to configure the device hardware features and memory protection. They are stored in a dedicated memory block. Refer to the option bytes section of the STM8S105xx datasheet for a detailed description.

STVP can be used to program the STM8S option bytes. It is part of the free software package provided by STMicroelectronics. This easy-to-use tool has a graphical interface allowing to read, write and verify the STM8 Flash programming memory, data EEPROM and option bytes.

The STVP executable is available from the Windows **start** menu.

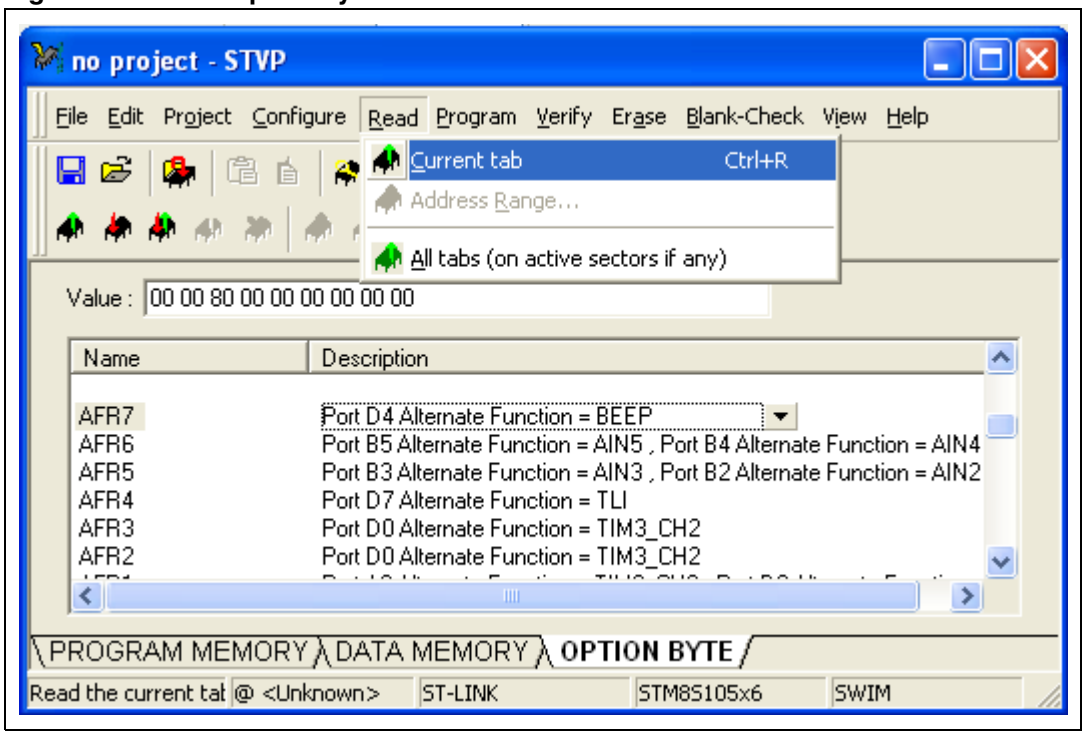
1. Open the STVP GUI and select **Configure>Configure ST Visual Programmer** from the main menu toolbar. The configuration dialog box opens.
2. In the **hardware** list, select ST-LINK as programming board.
3. In the **port** list, select USB as the host PC port to which the ST-LINK is connected. Only the ports that are compatible with the selected hardware are displayed.
4. in the **programming mode** list, select SWIM. The programming modes displayed depend on your programming equipment.
5. In the **device** list, select STM8S105x6 as the ST microcontroller you are going to program (see *Figure 18*).

**Figure 18. Select your MCU**




6. Select the **OPTION BYTE** tab, and click **Read >Current tab** in the menu toolbar. All the STM8S105 option bytes and their respective values are displayed in the window (see *Figure 19*).

**Figure 19. STVP option byte selection menu**




7. To program an option byte to a new value:
   a) Click the option byte description, and select the value.
   b) Click **Program>Current tab** in the main menu toolbar (see *Figure 20*).
   c) If the operation has completed successfully, the Output windows is displayed (see *Figure 21*).

**Figure 20. STVP option byte programming menu**

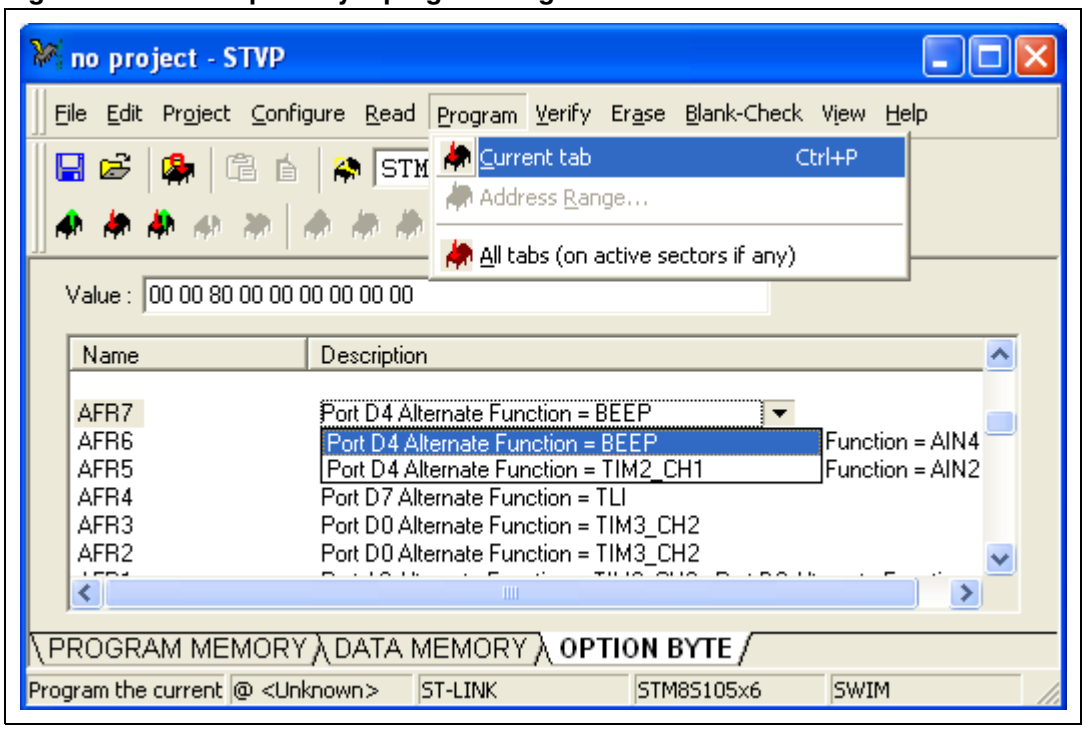
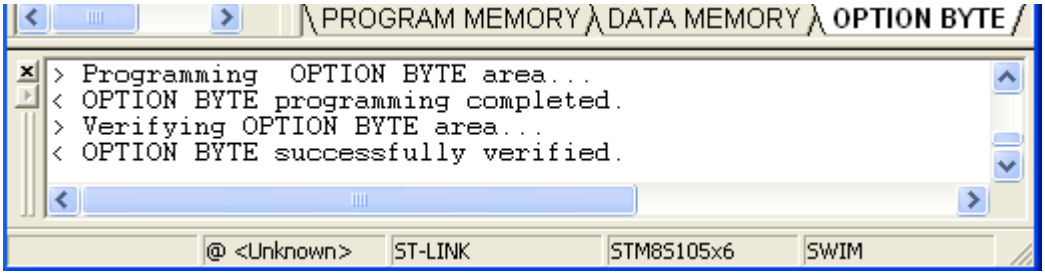



**Figure 21. STVP option byte programming message**



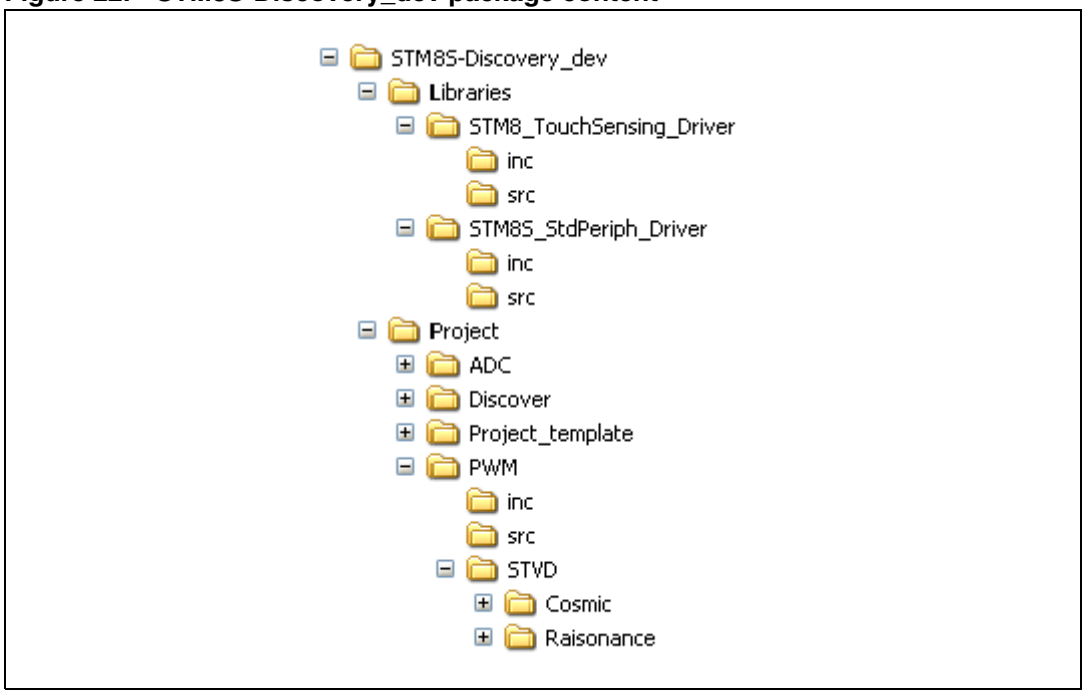
# 5 Building and running your STVD project

## 5.1 Description of the STM8S-DISCOVERY development package

ST provides a complete development package, the STM8S-Discovery_dev. It allows to easily get started with the STM8S-DISCOVERY by simplifying application code development and execution using the touch sensing or the standard peripherals drivers. This package can be downloaded together with the present user manual from http://www.st.com/stm8s-discovery.

The package is structured as follow:

**Figure 22. STM8S-Discovery_dev package content**




### 5.1.1 Libraries

The libraries are located in the **Libraries** directory.

- STM8_TouchSensing_Driver:
  This directory contains the source and header files of the STM8S touch sensing library
- STM8S_StdPeriph_Driver
  This directory contains the source and header files of the STM8S standard firmware library

**Warning: It is strongly recommended to upgrade the libraries as soon as new releases are available (see *Section 5.4*).**

### 5.1.2 Application code examples

The application code examples are located in the **Project** directory:

- Discover

  The Discover demonstration firmware comes preprogrammed in the STM8S-DISCOVERY. It makes LD1 LED blink at different speeds when the touch sensing key is pressed. This example uses the STM8S standard firmware library and the STM8 touch sensing library. For more details on the Discover application code implementation and library configuration, refer to user manual "*Adjustable LED blinking speed using STM8S-DISCOVERY touch sensing key*" (UM0833).

  The Discover application project is proposed only with the Cosmic compiler because the STM8 touch sensing library V1.3.0 included in the STM8S-Discovery_dev package is not yet compatible with the Raisonance compiler.

- PWM

  The PWM application code example is only based on the STM8S standard library. It explains how to use a timer to generate PWM signals. Refer to user manual "*Generating PWM signals using STM8S-DISCOVERY*" (UM0856) for details on the PWM application software code implementation and library configuration.

  The PWM project is compatible both with Cosmic and Raisonance compilers.

- ADC

  The ADC application code example explains how to use the ADC to convert the analog voltage delivered by a variable resistor. It does not use any library. Refer to user manual "*Read UM0848 "Displaying variable voltage on a LED bar*" (UM0848) for details on the ADC application software code implementation and project configuration.

  This example requires additional components to be mounted on the board.

  The ADC project is compatible both with Cosmic and Raisonance compiler.

- Project_template

  This directory contains a standard project template that should be used to start any application software development.

### 5.1.3 Project structure

Each project is structured as follows:

- **inc**

  This directory contains all the header files for your application, including the configuration files for the touch sensing library (*STM8_TSL_RC_Configuration.h*), and for the standard firmware library (*stm8s_conf.h*). These files are used to tailor the libraries for your application.

- **src**

  This directory contains all the source files dedicated to your application code, such as *main.c*, stm8s_it.c, and *stm8_interrupt_vector.c* (interrupt mapping file used only for Cosmic compiler).

- **STVD\Cosmic (or \Raisonance)**

  This directory is used to store STVD workspace files (for example *pwm.stw*). It includes the project and workspace for the Cosmic or Raisonance compiler, depending on the compiler you have selected.

## 5.2 Running the STM8S-Discovery examples

Before starting developing your own application, run one of the examples provided with the STM8S-DISCOVERY development package to check if your STM8S-DISCOVERY is correctly configured.

Follow the steps below:

1. Install the development toolchain (see *Section 2: Installing the development toolchain*)
   - Cosmic or Raisonance compiler
   - STVD
2. Download the STM8S-Discovery_dev.zip file and extract the directory **STM8S-Discovery_dev** on your PC (see *Section 5.1: Description of the STM8S-DISCOVERY development package*)
3. Run STVD

   From **START>Programs>ST Toolset>Development Tools>ST Visual Develop**

   a) Select **File -> Open Workspace**

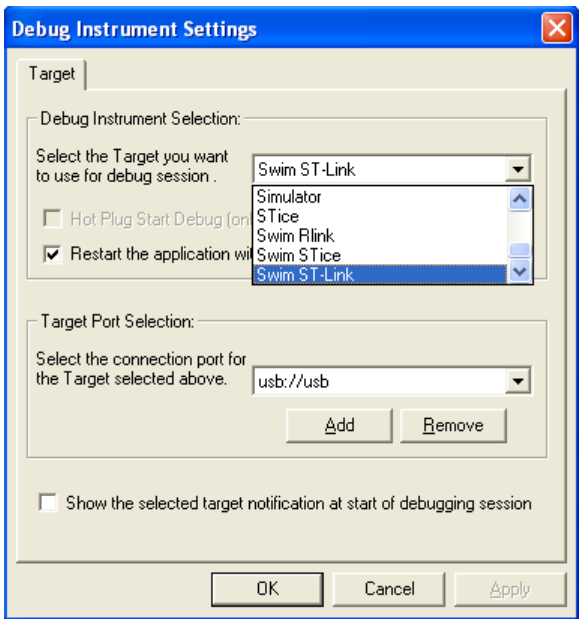
   b) Browse for the project file .stw of the example you intend to run according to the compiler you have installed:

   - Discover example: *STM8S-Discovery_dev\Project\Discover\STVD\Cosmic\Discover.stw*
   - PWM example: *STM8S-Discovery_dev\Project\PWM\STVD\Cosmic\PWM.stw* or *STM8S-Discovery_dev\Project\PWM\STVD\Raisonance\PWM.stw*
   - ADC example: *STM8S-Discovery_dev\Project\ADC\STVD\Cosmic\ADC.stw* or *STM8S-Discovery_dev\Project\ADC\STVD\Raisonance\ADC.stw*
4. Configures STVD to use ST-LINK as debug instrument (see *Figure 23*):

   a) Click **Debug instrument -> Target Settings**.

   b) Select target **Swim ST-LINK** in the target list.
5. Click **Build -> Rebuild All** to build your application.
6. Clicking **Debug -> Start Debugging** to start the debug session.
7. Click **Debug -> Run** to run you application.
8. The application code example you have selected is now running. Check if it operates in compliance with what is described in the associated user manual. You can also use STVD debug capabilities to run the code step by step and monitor the variables (see *Section 6.2: Debugging your application*).

**Figure 23. Selecting the debug instrument**




## 5.3 Creating your STVD project

This section explains step by step how to create your own application project. The touch sensing and the firmware libraries can be used or not according to the kind of application code to develop.

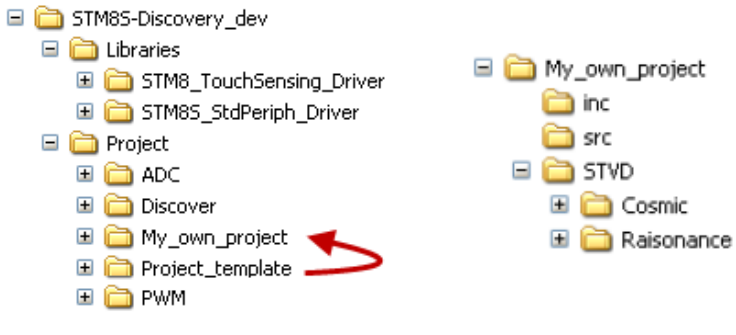
All projects must be created starting from STM8S-Discovery_dev development package (see *Section 5.1: Description of the STM8S-DISCOVERY development package*).

### 5.3.1 Creating your project structure

The best way to proceed is to start from the **Project_template** directory:

1. Extract the content of STM8S-Discovery_dev.zip file on your PC.
2. Copy the **Project_template** directory and rename it **My_own_project** (see *Figure 24*). Your STVD project comes already structured thus simplifying the project creation.

**Figure 24. Creating your project structure**

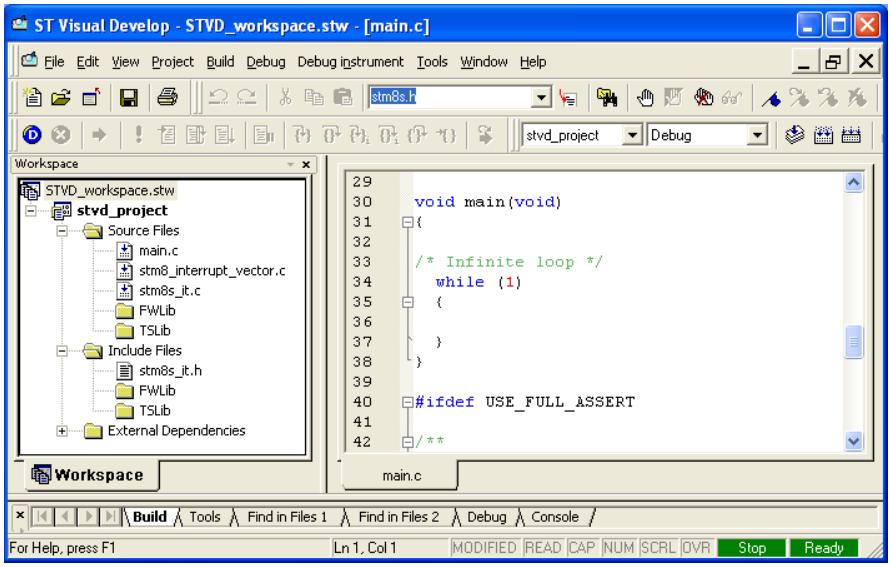


### 5.3.2 Creating your STVD project

1. Open STVD and click **File>Open Workspace**
2. Browse for the **Workspace location** and select **STVD_workspace.stw** in the **Cosmic** or **Raisonance** directory depending on the compiler you intend to use:
   - For Cosmic, select **\STM8S-DISCOVERY_dev\Project\My_own_project\STVD\Cosmic\STVD_workspace.stw**.
   - for Raisonance, select **\STM8S-DISCOVERY_dev\Project\My_own_project\STVD\Raisonance\STVD_workspace.stw**.

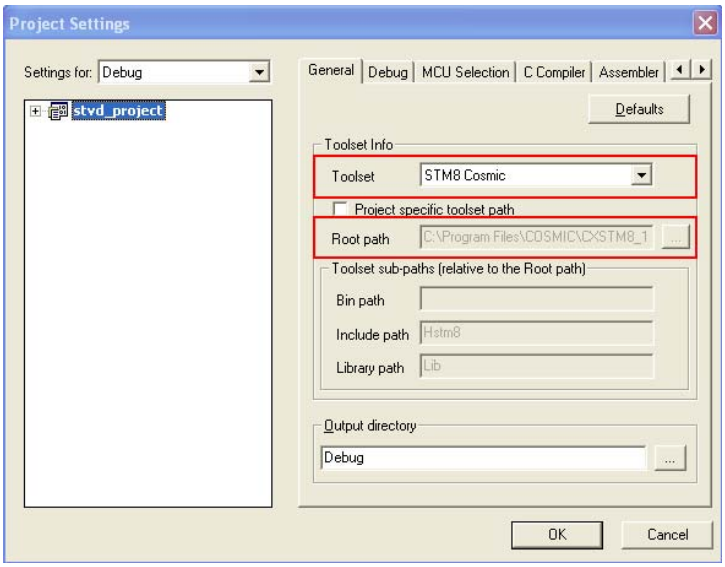
   **stvd_project** appears in the STVD workspace window as shown in *Figure 25*. Its structure is slightly different from the structure of **My_own_project** directory in **STM8S-Discovery_dev** package:
   - **Source Files** has the same content as **\STM8S-DISCOVERY_dev\Project\My_own_project\STVD\src**.
   - **Source Files\FWLib** contains the STM8S standard library source files.
   - **Source Files\TSLib** contains the STM8S touch sensing library source files.
   - **Include Files** contains **My_own_project** include files.
   - **Include Files\FWLib** contains the STM8S standard library include files.
   - **Include Files\TSLib** contains the STM8S touch sensing library include files.

**Figure 25. Creating the STVD workspace for your project**



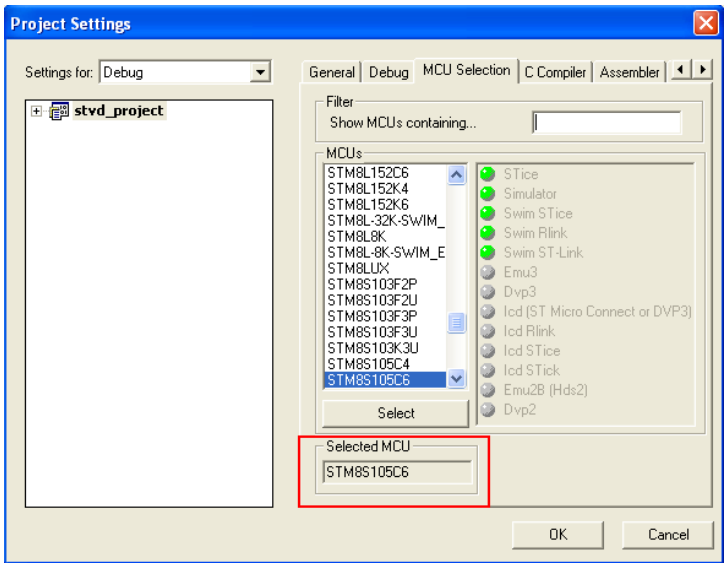

3. Check that the compiler (STM8 Cosmic or Raisonance) and its path are correct by selecting **Project>settings** from the STVD main menu (see *Figure 26*).

**Figure 26. Checking the selected compiler**



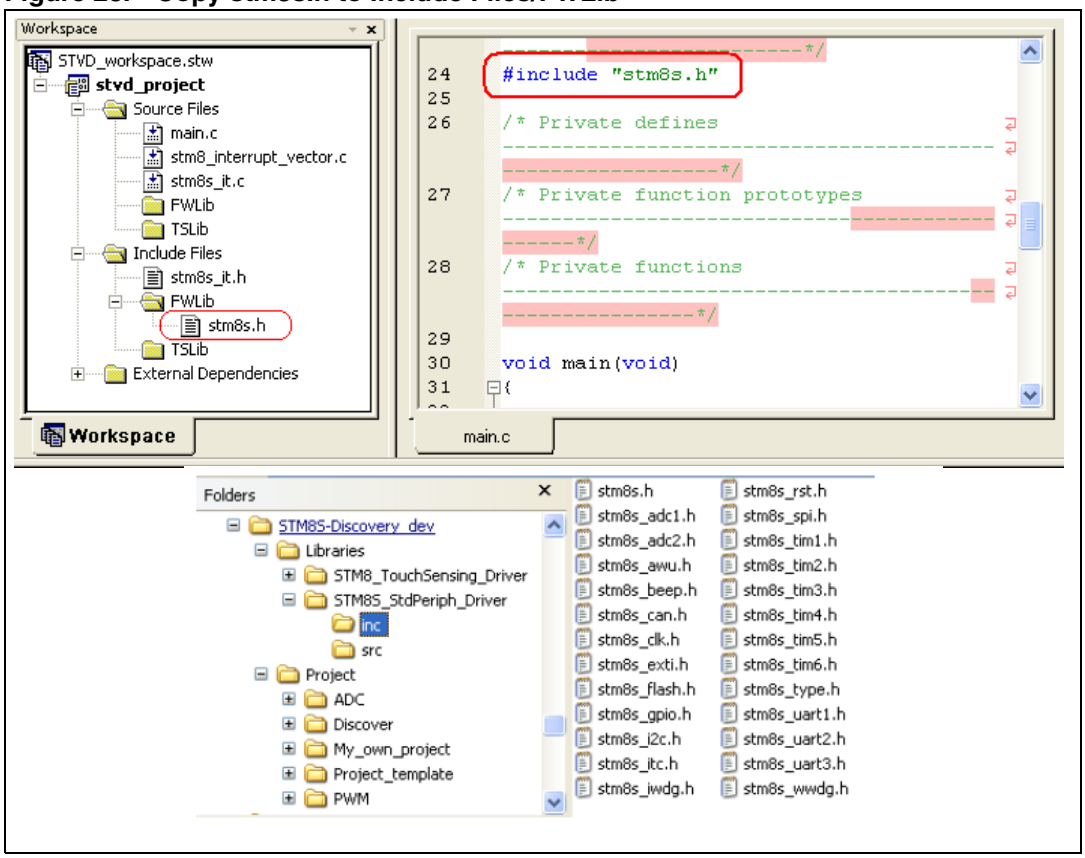

4. Go to **Projects>settings>MCU selection**, and select **STM8S105C6** from the MCUs list. Make sure that MCU selected is displayed in the selected MCU field before clicking OK (see *Figure 27*).

**Figure 27. Select the MCU**



5. Configure STVD to use ST-LINK as debug instrument (see *Figure 23*):
   a) Click **Debug instrument -> Target Settings**
   b) Select target **Swim ST-Link** in the target list
6. Copy *stm8s.h* to the **Include Files/FWLib** directory of your STVD workspace (see *Figure 28*):
   – Right click on the **FWLib** directory under **stvd_project**
   – Select **Add Files to Folder**
   – Browse the *sm8s.h* file location in **\STM8S-DISCOVERY_dev\Libraries\ STM8S_StdPeriph_Driver\inc**.
7. Make sure `#include <stm8s.h>` is present of your *main.c* file.

**Figure 28. Copy *stm8s.h* to Include Files/FWLib**




## 5.3.3 Linking the libraries to your STVD project

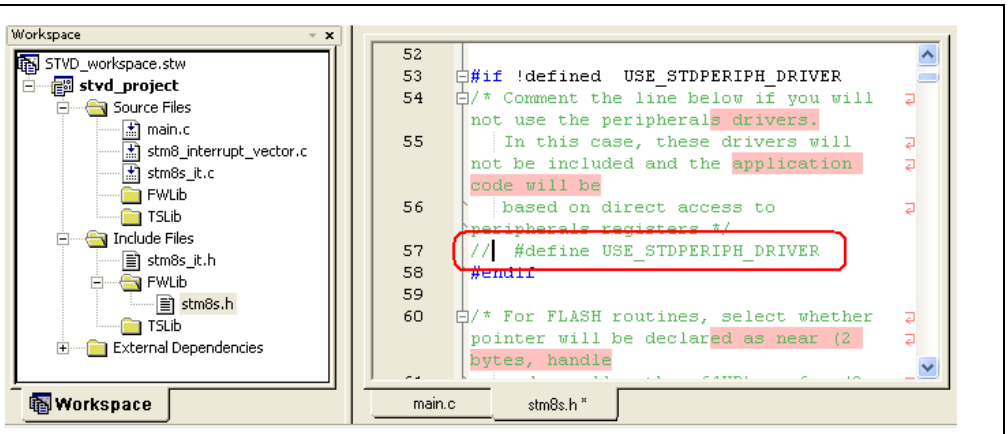
### No library linked to your STVD project

If your application does not require any library, follow the instructions below:

1. Edit *stm8s.h* and comment `#define USE_STDPERIPH_DRIVER` to be able to use the register structures, memory mapping, and constant definitions for each peripheral.

   For an example of application code that does not use any library, refer to the ADC project of the STM8S-DISCOVERY development package.

**Figure 29. No libraries linked**




## Linking the STM8S standard firmware library to your STVD project

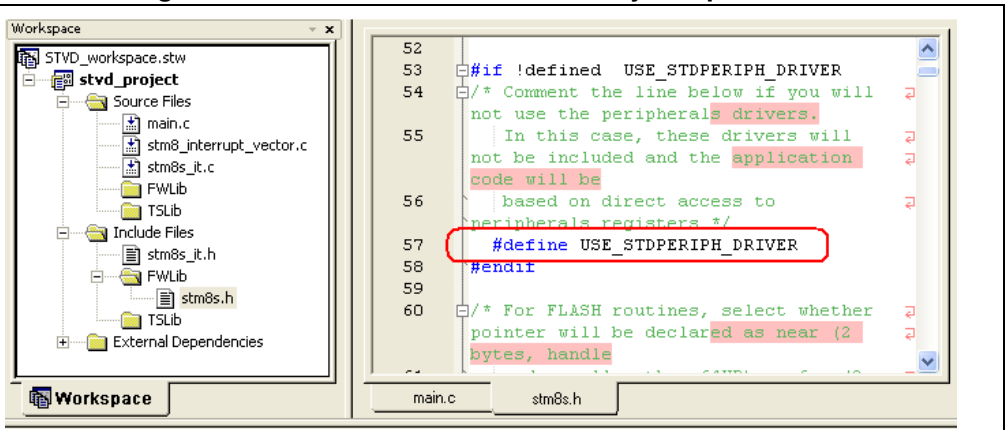
This section explains step by step how to link the STM8S standard firmware library to your STVD project.

Prior to linking the library to your project, check that the STM8S library is up-to-date. If it is not, download the latest release (see *Section 5.4: Updating the libraries*).

Follow the steps described below to link the STM8S standard firmware library to your STVD project:

1. Edit *stm8s.h* and uncomment `#define USE_STDPERIPH_DRIVER` (see *Figure 30*).

**Figure 30. Linking the STM8S standard firmware library - step 1**




2. Copy *stm8s_conf.h* from the **My_own_project\inc** directory of **STM8S-DISCOVERY_dev** to the **Include File** root of your STVD project.

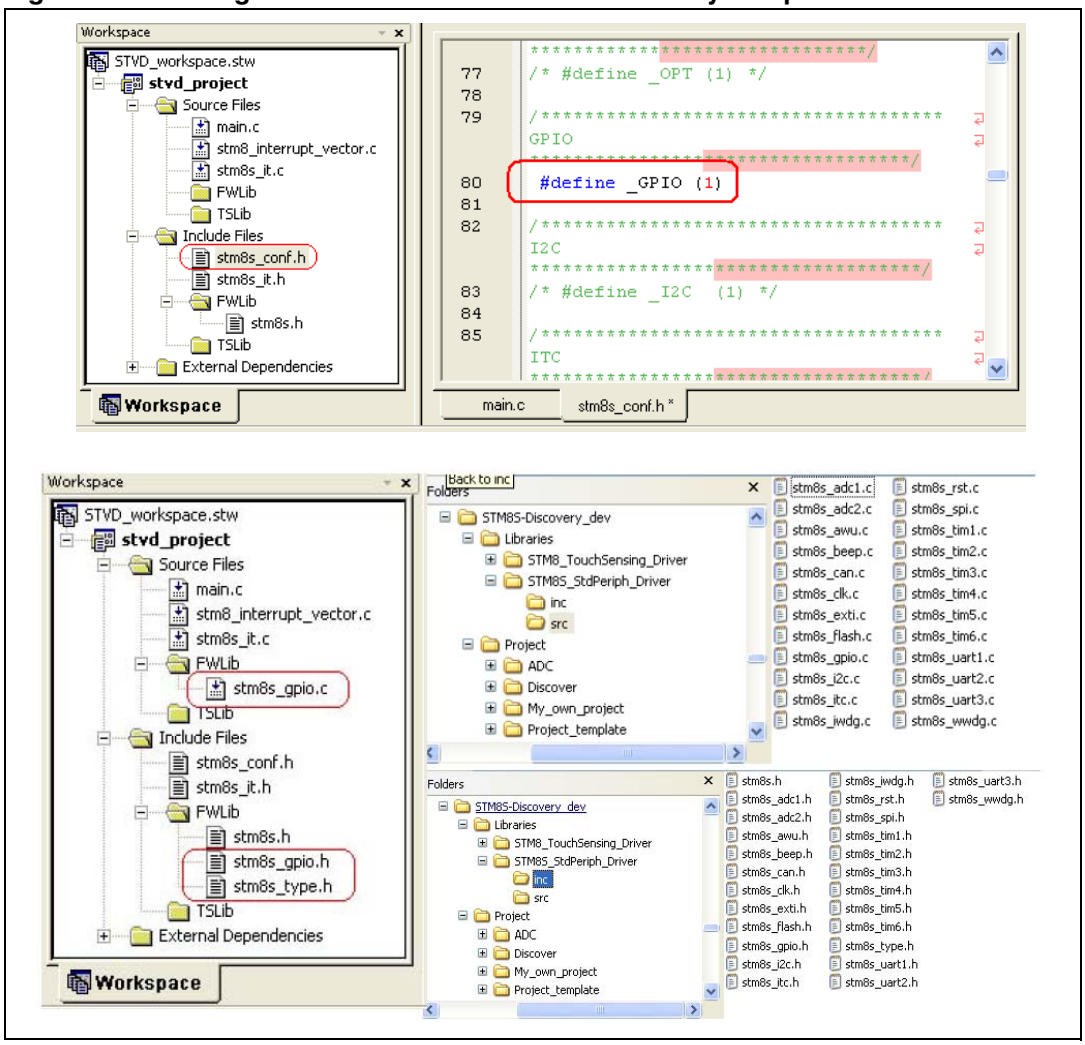
   In *stm8s_conf.h*, uncomment the define statements corresponding to the peripherals used by your application.

   *stm8s_type.h* plus the .c and .h files corresponding to the peripherals which define statement are uncommented must be copied from **STM8S-DISCOVERY_dev\Libraries\STM8S_StdPeriph_Driver\inc** and **STM8S-**

**DISCOVERY_dev\Libraries\STM8S_StdPeriph_Driver\src** into **Include Files\FWLib** and **Source Files\FWLib** of STVD project, respectively.

You can also decide to copy the whole drivers (.c and .h files) from the library to your STVD project.

**Figure 31. Linking the STM8S standard firmware library - step 2**



### Linking the STM8 touch sensing library to your STVD project

This section explains step by step how to link the STM8 touch sensing library to your STVD project.

Prior to linking the library to your project, check that the library is up-to-date. If it is not, download the latest release (see *Section 5.4: Updating the libraries*).

Follow the steps below to link the touch sensing library to your STVD project:

1. Configure the library according to the number of touchkeys (up to 24), sliders (up to 2) and wheels (up to 2) used in your application:
   a) Configure *STM8_TSL_RC_Configuration.h*

      To use only the touch sensing key TS1, reuse the configuration file from the **Discover** project by copying *STM8_TSL_RC_Configuration.h* from **STM8S-DISCOVERY_dev\Project\Discover\inc** to your project folder **STM8S-DISCOVERY_dev\Project\My_own_project\inc**.

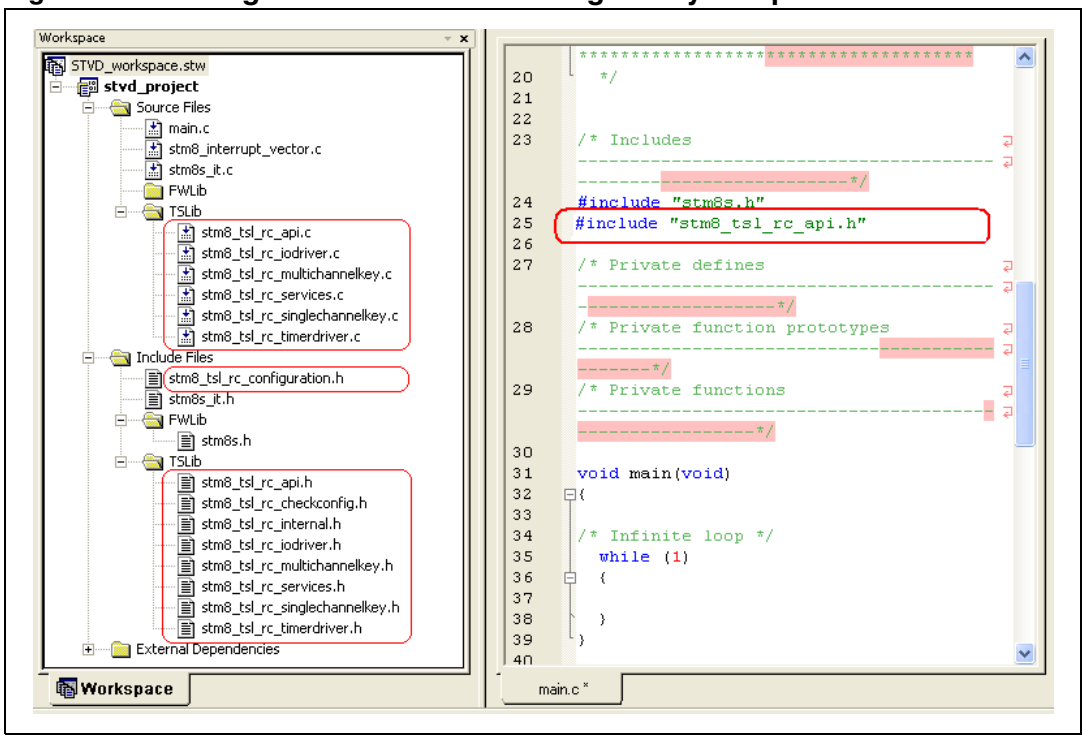
      Otherwise, copy the original configuration file, *STM8_TSL_RC_Configuration_TOADAPT.h*, available in **STM8S-DISCOVERY_dev\Libraries\STM8_TouchSensing_Driver\inc** to your project folder **STM8S-DISCOVERY_dev\Project\My_own_project\inc.** Rename it *STM8_TSL_RC_Configuration.h*. Set the parameters in compliance with your application hardware (refer to the touch sensing library online help available in the installation directory).

   b) Copy *STM8_TSL_RC_Configuration.h* to the root of the **Include Files** directory in the STVD workspace panel.

   c) All the sources and header files (.c and .h) except for the *STM8_TSL_RC_Configuration_TOADAPT.h* (see above) have to be copied from **STM8SDISCOVERY_dev\Libraries\STM8S_TouchSensing_Driver\src** and **STM8SDISCOVERY_dev\Libraries\STM8S_TouchSensing_Driver\inc** to **Source Files\TSLib** and **Include Files\TSLib,** respectively.

   d) Add the `#include <stm8_tsl_rc_api.h>` statement in your *main.c* file.

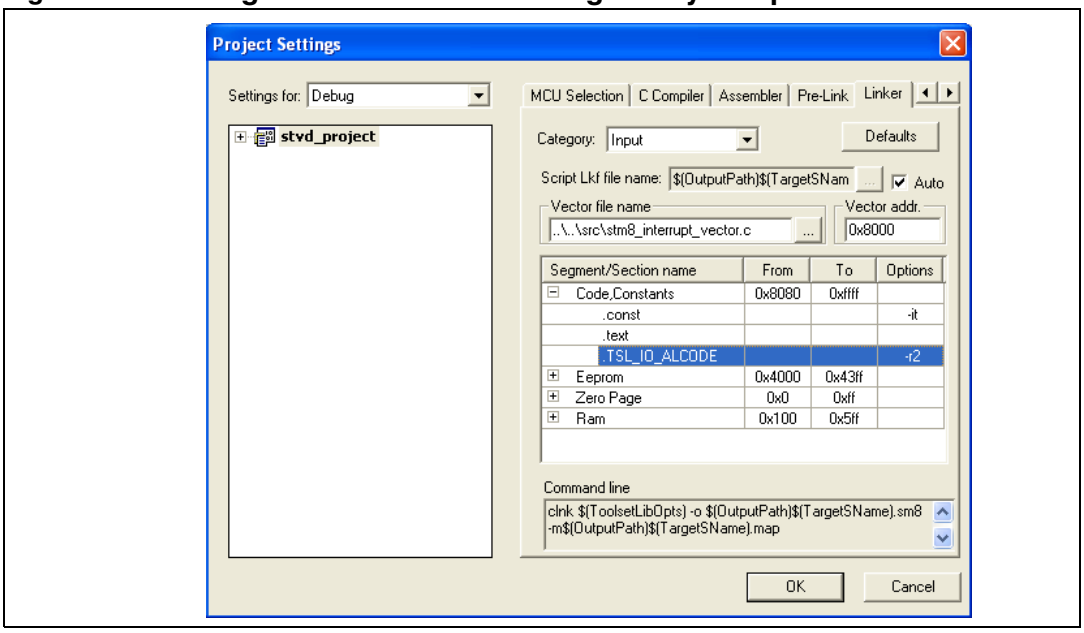
**Figure 32. Linking the STM8 touch sensing library - step 1**




2. Some functions must be aligned to even memory addresses (see *Figure 33*):
   a) Click **project>setting** from the STVD main menu.
   b) Select the **linker** tab and then select the **input category**.
   c) Under **Segment/Section name,** add **TSL_IO_ALCODE** in the **code,Constants** section and assign the **-r2** option to it.

*Note: A specific main.c structure is required to correctly use the touch sensing features. This structure is already implemented in the Discover project. Please refer to Section 3.2: STM8S touch sensing library for more information, and to user manual "Adjustable LED blinking speed using STM8S-DISCOVERY touch sensing key" (UM0833) for an example.*

**Figure 33. Linking the STM8 touch sensing library - step 2**




## 5.4 Updating the libraries

### 5.4.1 Updating the STM8S standard firmware library

To update the STM8S standard firmware library:

1. Download the new release from http://www.st.com/mcu.
2. Copy the **inc** and **src** directories in **\"STM8S-DISCOVERY_dev"\Libraries\STM8S_StdPeriph_Driver**.

### 5.4.2 Updating the STM8 touch sensing library

To update the STM8 touch sensing library:

1. Download the new release from http://www.st.com/mcu.
2. Copy the **inc** and **src** directories in **\"STM8S-Discovery_My_own_project"\Libraries\STM8_TouchSensing_Driver**.

# 6 Building, debugging and running your application

Once your have developed your application code, created your workspace environment, and launched STVD, you can start building, debugging and programming your application to the target microcontroller.

## 6.1 Building your application

Once your project is created, the build context is enabled by default. It allows to access all the commands required to set up, customize, and build your application.

The build configuration is available by selecting ***Build>Configurations*** from the STVP main menu toolbar. It allows to change the application building settings.

Two preset configurations are available in STVD:

- **Debug**

  This configuration creates a version of your application that allows using all the STVD advanced debugging features. When using this configuration, output files are saved in the **Debug** directory in your **workspace** directory.

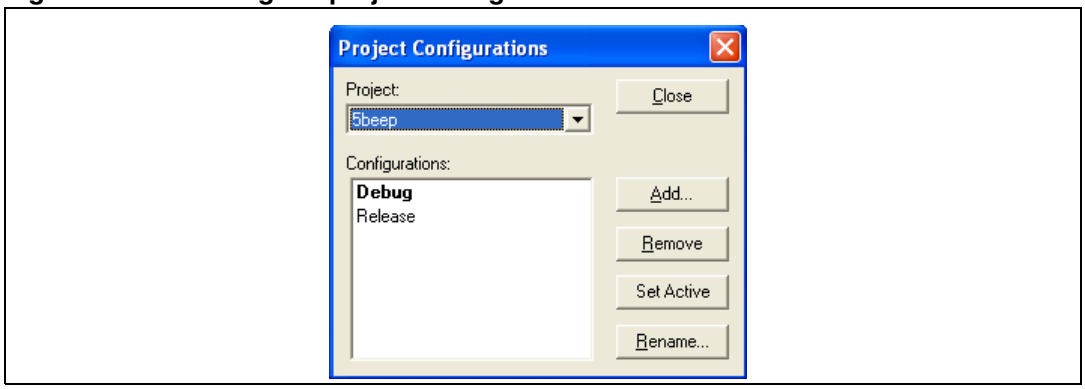
- **Release**

  This configuration creates a version of your application that uses the default optimization for your toolset. This version of your application is ready to be programmed to your target microcontroller.

Follow the sequence below to build a project:

1. Ensure that the **Debug** configuration is selected by clicking **Build>Configurations**. If the **Debug** configuration is not selected, highlight it and click **Set Active** (see *Figure 34*).

**Figure 34. Selecting the project configuration**




2. Once the building options are correctly configured, configure your project settings:
   a) Select ***Project > Settings*** from the STVD main menu toolbar.
   b) The **Project settings** window opens and displays all the options of your toolset compiler, assembler, and linker. You can then customize these options. For more

information on the available options for your toolset, refers to the STVD online help available by clicking **Help** in the STVD main menu toolbar.

3. Once your MCU is selected (***Project > Settings > MCU Selection***) and your building options are configured, use the **Build** menu to build and rebuild your application, or compile your source files (see *Figure 35*).
4. The build command lines and possible warnings and/or errors are displayed in the **build** tab of the **Output** window (see *Figure 36*).

**Figure 35. Building your project**

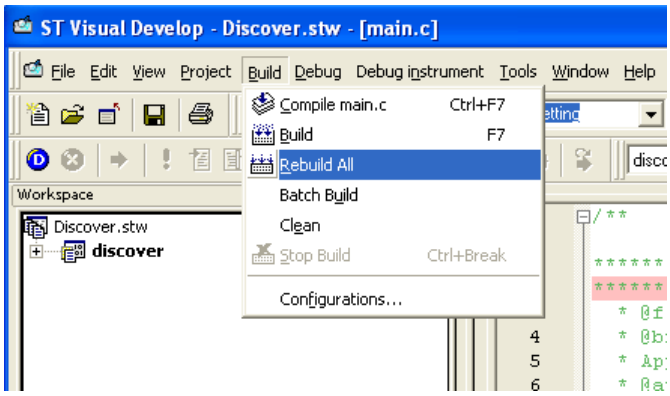
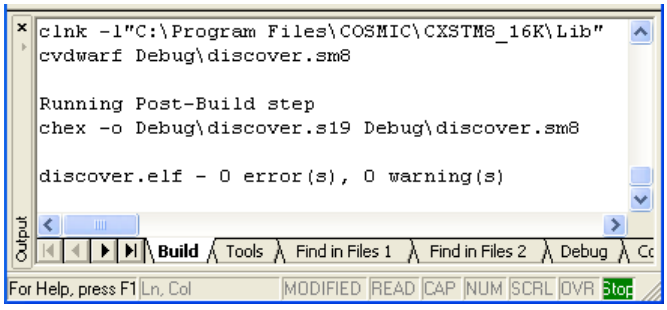



**Figure 36. Building successful message**

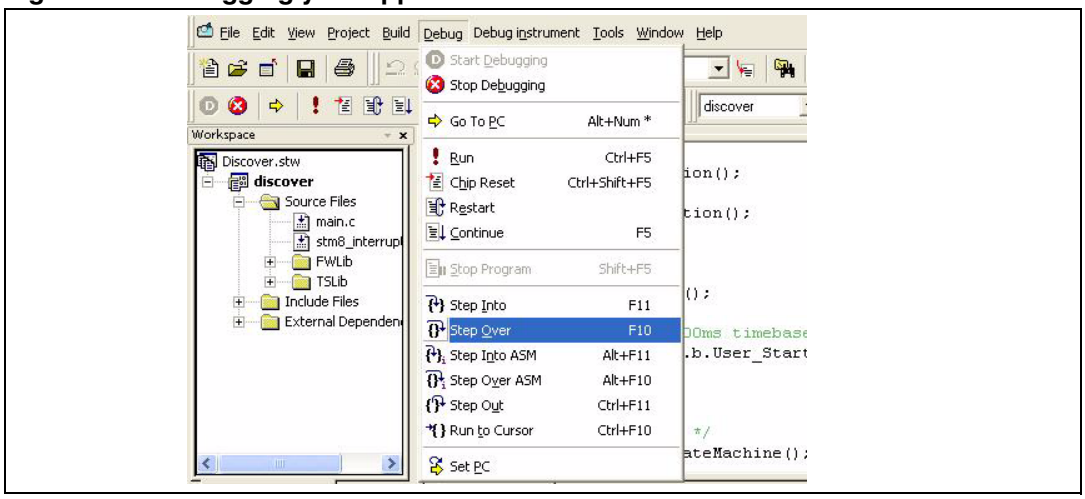

clnk -l"C:\Program Files\COSMIC\CXSTM8_16K\Lib"
cvdwarf Debug\discover.sm8
Running Post-Build step
chex -o Debug\discover.s19 Debug\discover.sm8
discover.elf - 0 error(s), 0 warning(s)
Output
Build Tools Find in Files 1 Find in Files 2 Debug
For Help, press F1 Ln, Col MODIFIED READ CAP NUM SCRL OVR

## 6.2 Debugging your application

Once your application has been successfully built, access the **Debug context** from the STVD main menu to access the debugging features that are supported by your debugging instrument. The debug instrument must be selected before starting debugging.

1. Select ***Debug Instrument > Target settings*** from the STVD main menu toolbar.
2. Select **Swim ST-LINK**, which is your debug instrument (see *Figure 23*).
3. Click **Apply** to confirm and **OK** to close the window.
4. Select ***Debug > Start Debugging*** to start your debug session and access the debug context. STVD then connects to your debug instrument, loads the code into the microcontroller Flash memory, and provides access to the debug commands and menus. You can now start debugging your application (see *Figure 37*).

**Figure 37. Debugging your application**




Several debug windows are available in STVD. You can access them by selecting **View menu** when a debug session is ongoing.

One of the most useful windows is the **Peripheral registers window.** It can be used to display the content of the STM8S105C6 peripheral registers during the debug session. (see *Figure 38*).

The **Watch window** displays the values the code variables. Just drag and drop a variable from your code into the **Watch window**.

The information displayed in the debug windows are refreshed when the program is stopped (for example by a break point). (see *Figure 39*)

**Figure 38. Peripheral registers window**

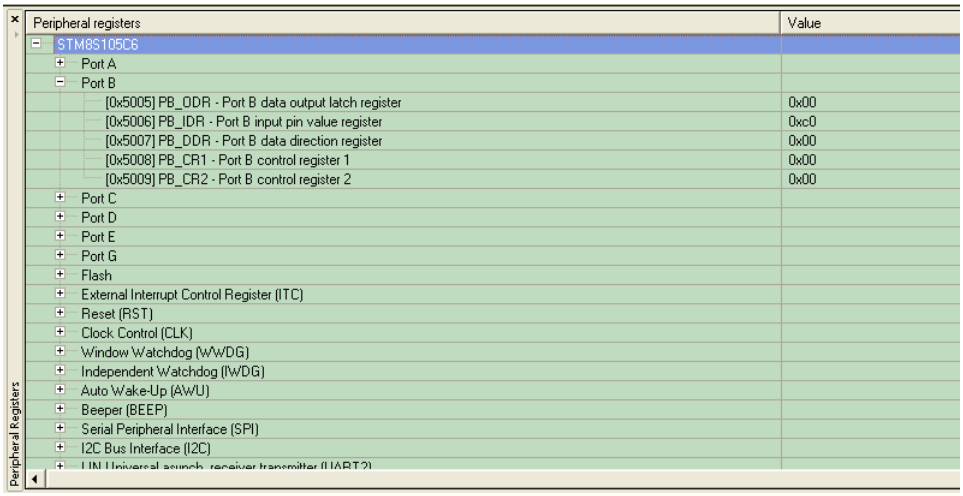
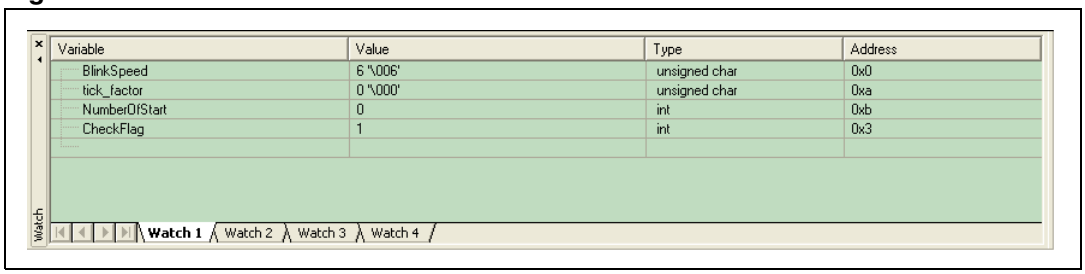


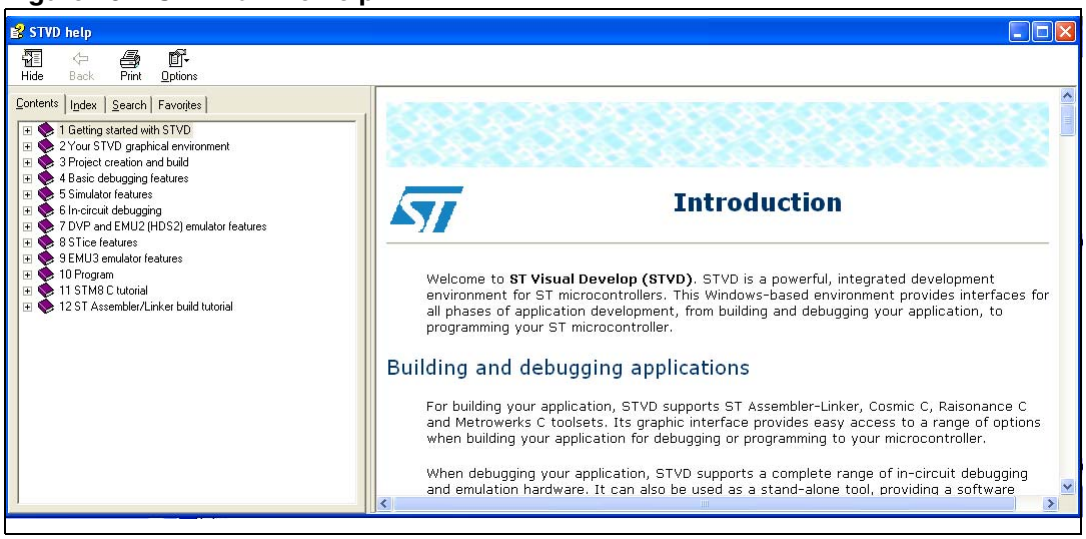
**Figure 39. Watch window**




## 6.3 Online help

For more information on building and debugging features, refer to embedded STVD online help page (**Help>Help Home Page)**.

**Figure 40. STVD online help**


STVD help
Hide
Back
Print
Options
Contents
Index
Search
Favorites
1 Getting started with STVD
2 Your STVD graphical environment
3 Project creation and build
4 Basic debugging features
5 Simulator features
6 In-circuit debugging
7 DVP and EMU2 (HDS2) emulator features
8 STice features
9 EMU3 emulator features
10 Program
11 STM8 C tutorial
12 ST Assembler/Linker build tutorial
Introduction
Welcome to ST Visual Develop (STVD). STVD is a powerful, integrated development environment for ST microcontrollers. This Windows-based environment provides interfaces for all phases of application development, from building and debugging your application, to programming your ST microcontroller.
Building and debugging applications
For building your application, STVD supports ST Assembler-Linker, Cosmic C, Raisonance C and Metrowerks C toolsets. Its graphic interface provides easy access to a range of options when building your application for debugging or programming to your microcontroller.
When debugging your application, STVD supports a complete range of in-circuit debugging

# 7 Revision history

**Table 2. Document revision history**

| Date | Revision | Changes |
|---|---|---|
| 04-Nov-2009 | 1 | Initial release. |
| 08-Feb-2010 | 2 | Updated list of reference documents on cover page.<br>Modified url of STM8S-DISCOVERY page.<br>Added warning in *Section 3.2: STM8S touch sensing library*<br>Reworked *Section 5: Building and running your STVD project*. |
| 04-Jan-2011 | 3 | Replaced http://www.st.com/stonline/products/support/micro/files/sttoolset.exe by http://www.st.com. |

**Please Read Carefully:**

Information in this document is provided solely in connection with ST products. STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, modifications or improvements, to this document, and the products and services described herein at any time, without notice.

All ST products are sold pursuant to ST’s terms and conditions of sale.

Purchasers are solely responsible for the choice, selection and use of the ST products and services described herein, and ST assumes no liability whatsoever relating to the choice, selection or use of the ST products and services described herein.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted under this document. If any part of this document refers to any third party products or services it shall not be deemed a license grant by ST for the use of such third party products or services, or any intellectual property contained therein or considered as a warranty covering the use in any manner whatsoever of such third party products or services or any intellectual property contained therein.

**UNLESS OTHERWISE SET FORTH IN ST’S TERMS AND CONDITIONS OF SALE ST DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY WITH RESPECT TO THE USE AND/OR SALE OF ST PRODUCTS INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE (AND THEIR EQUIVALENTS UNDER THE LAWS OF ANY JURISDICTION), OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.**

**UNLESS EXPRESSLY APPROVED IN WRITING BY AN AUTHORIZED ST REPRESENTATIVE, ST PRODUCTS ARE NOT RECOMMENDED, AUTHORIZED OR WARRANTED FOR USE IN MILITARY, AIR CRAFT, SPACE, LIFE SAVING, OR LIFE SUSTAINING APPLICATIONS, NOR IN PRODUCTS OR SYSTEMS WHERE FAILURE OR MALFUNCTION MAY RESULT IN PERSONAL INJURY, DEATH, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE. ST PRODUCTS WHICH ARE NOT SPECIFIED AS "AUTOMOTIVE GRADE" MAY ONLY BE USED IN AUTOMOTIVE APPLICATIONS AT USER’S OWN RISK.**

Resale of ST products with provisions different from the statements and/or technical features set forth in this document shall immediately void any warranty granted by ST for the ST product or service described herein and shall not create or extend in any manner whatsoever, any liability of ST.

ST and the ST logo are trademarks or registered trademarks of ST in various countries.

Information in this document supersedes and replaces all information previously supplied.

The ST logo is a registered trademark of STMicroelectronics. All other names are the property of their respective owners.

© 2011 STMicroelectronics - All rights reserved

STMicroelectronics group of companies

Australia - Belgium - Brazil - Canada - China - Czech Republic - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan - Malaysia - Malta - Morocco - Philippines - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States of America

**www.st.com**